



DATA SCIENCE FOR ENGINEERS

Prof. Ragunathan Rengasamy
Prof. Shankar Narasimhan
Computer Science and Engineering
IIT Madras



INDEX

S. No	Topic	Page No.
<i>Week 1</i>		
1	Data Science for Engineers Course Philosophy and Expectation	1
2	Introduction to R	6
3	Introduction to R (Continued)	18
4	Variables and Datatypes in R	27
5	Data frames	39
6	Recasting and Joining of Dataframes	50
7	Arithmetic, Logical and Matrix Operations in R	68
8	Advanced Programming in R : Functions	90
9	Advanced Programming in R : Functions (Continued)	99
10	Control Structures	109
11	Data Visualization in R Basic Graphics	118
<i>Week 2</i>		
12	Linear Algebra for Data science	126
13	Solving Linear Equations	148
14	Solving Linear Equations (Continued)	161
15	Linear Algebra - Distance,Hyperplanes and Halfspaces,Eigenvalues,Eigenvectors	174
16	Linear Algebra - Distance,Hyperplanes and Halfspaces,Eigenvalues,Eigenvectors (Continued 1)	190
17	Linear Algebra - Distance,Hyperplanes and Halfspaces,Eigenvalues,Eigenvectors (Continued 2)	202
18	Linear Algebra - Distance,Hyperplanes and Halfspaces,Eigenvalues,Eigenvectors (Continued 3)	217
<i>Week 3</i>		
19	Statistical Modelling	228
20	Random Variables and Probability Mass/Density Functions	241
21	Sample Statistics	256

22	Hypotheses Testing	272
----	--------------------	-----

Week 4

23	Optimization for Data Science	289
24	Unconstrained Multivariate Optimization	306
25	Unconstrained Multivariate Optimization (Continued)	317
26	Gradient (Steepest) Descent (OR) Learning Rule	324

Week 5

27	Multivariate Optimization With Equality Constraints	333
28	Multivariate Optimization With Inequality Constraints	343
29	Introduction to Data Science	359
30	Solving Data Analysis Problems - A Guided Thought Process	374

Week 6

31	Module : Predictive Modelling	386
32	Linear Regression	402
33	Model Assessment	422
34	Diagnostics to Improve Linear Model Fit	436
35	Simple Linear Regression Model Building	451
36	Simple Linear Regression Model Assessment	461
37	Simple Linear Regression Model Assessment (Continued)	468
38	Muliple Linear Regression	479

Week 7

39	Cross Validation	498
40	Multiple Linear Regression Modelling Building and Selection	512
41	Classification	524
42	Logisitic Regression	530
43	Logisitic Regression (Continued)	542
44	Performance Measures	551

45	Logisitic Regression Implementation in R	565
----	--	-----

Week 8

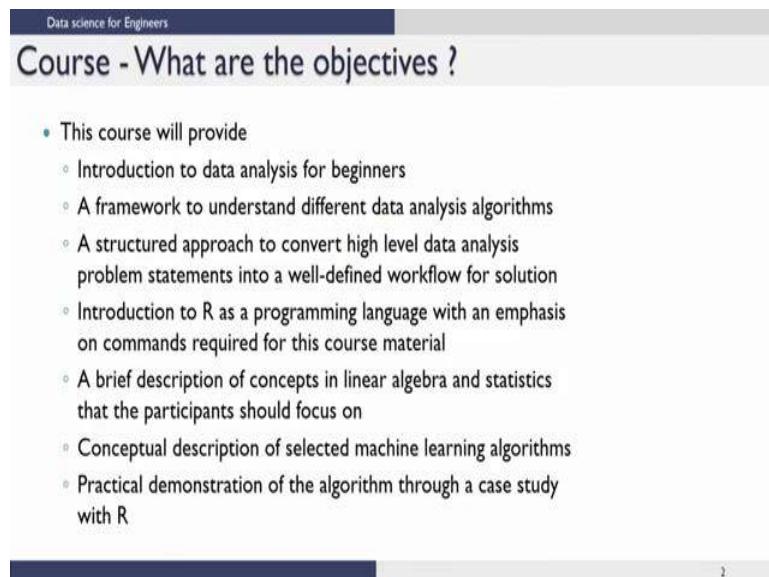
46	K - Nearest Neighbors (kNN)	584
47	K - Nearest Neighbors implementation in R	596
48	K - means Clustering	614
49	K - means Implementation in R	626
50	Data Science for Engineers - Summary	642

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 01
Data science for engineers - Course philosophy and expectation

Welcome, to this course on Data Science for Engineers. My name is Raghunathan Rengaswamy. I am a professor in the Indian Institute of Technology at Madras. I will be teaching this course with my colleague professor Shankar Narasimhan also from IIT, Madras. The, teaching assistants for this course are Doctor Hemanth Kumar Tanneru and Miss Shweta Shridhar. In this very brief video, I am going to talk about the course philosophy and the expectations that you could have from this course. Let us start with the objectives of the course.

(Refer Slide Time: 00:59)



Data science for Engineers
Course - What are the objectives ?

- This course will provide
 - Introduction to data analysis for beginners
 - A framework to understand different data analysis algorithms
 - A structured approach to convert high level data analysis problem statements into a well-defined workflow for solution
 - Introduction to R as a programming language with an emphasis on commands required for this course material
 - A brief description of concepts in linear algebra and statistics that the participants should focus on
 - Conceptual description of selected machine learning algorithms
 - Practical demonstration of the algorithm through a case study with R

First off, I want to say, this is the first course on data analysis for beginners. So, this is for people who want to learn data analytics who have not been practicing it for a long time and so on. However, while we say this is a data analysis course for beginners, it would still be a substantial amount of information substantial amount of mathematical concepts and more conceptual ideas that we will have to teach.

So, while it is an introduction course it is still significant amount of effort and learning that that we expect the participants to get out of this course when we talk about data

analytics, there are several algorithms that one could use for doing analytics. So, as part of this course, we will try as much as possible whenever appropriate to explain all the concepts in terms of the data science problems that one might use them to solve. So, in that sense, you try to give you a framework to understand different data analysis problems and algorithms and we will also as much as possible try and provide a structured approach to convert high level data analytic problem statements into what we call as well defined workflow for solutions. So, you take a problem statement and then see how you can break it down into smaller components and solve using an appropriate algorithm.

So, these are at a conceptual level what you would expect the participants to take out of this course. For teaching data analytics or data science it is imperative that you do coding in a particular language there are many possibilities here as far as this course is concerned we are going to use R as a programming language. So, as part of this course R will also be introduced and the emphasis here will be on the aspects of R that are more critical for what you learn in this course. So, in other words commands that are required for this course material will be dealt in sufficient detail.

So, that is as far as a programming language is concerned for learning data science. In terms of the mathematics behind all of this we will describe important concepts in linear algebra that we think are critical for good understanding of machine learning and data science algorithms we will teach those and we will also teach statistics that are relevant for data science. Other than this will also have modules on optimization ideas and optimization that are directly relevant in machine learning algorithms, we will also provide conceptual and descriptions that are easy to understand for selected machine learning algorithms and whenever we teach a machine learning algorithm we will also follow it up with another lecture where the practical implementation of an algorithm for a problem statement is demonstrated and that demonstration would take place and we will use R as the programming platform.

(Refer Slide Time: 04:48)

Course- What it is not?

- This course is not for practitioners of advanced data analysis
- This course is not about big data implementation concepts such as map reduce, hadoop frameworks and so on
- Not an in-depth R programming course
- Only a selected few machine learning techniques that are most relevant for a beginner are taught

3

While we talk about what the objectives of this course are it is also a good idea to understand what this course is not about. As I mentioned already if you are a very advanced data analysis practitioner then there are other courses which are at more advanced levels that are relevant, this course is at a basic level for someone to get into this field of data science. We will be teaching a course on machine learning later which might be more appropriate for people of this category. This course is also not about big data per se and we are not going to cover big data concepts such as map reduce, hadoop frameworks and so on.

This course is more about the mathematical side of the data analytics, so, we are going to focus more on the algorithms and what are the fundamental ideas that underlie these algorithms. While we will use R as a programming platform this is not an in depth R programming course where we teach you very sophisticated programming techniques in R the R programming platform will be used in as much as it is important for us to teach the underlying data science algorithms.

Now, there are a wide variety of machine learning techniques there are a number of techniques that could be used and in a nine week course we have to pick the techniques that are most relevant, not only that since we think of this as a first course in data science. We also have to spend enough time covering the fundamental topics of linear algebra statistics and optimization from a data science perspective. So, that takes quite a few weeks of lecture. So, we are going to pick a few machine techniques which we believe are the most relevant for a beginner.

So, you understand the basic ideas in data science you get a fundamental grounding on the math principles that you need to learn and then you put all of this together in some machine learning technique. So, you understand some machine learning techniques where all of these ideas are used and we have picked these techniques in such a way that you can understand data science better and also use these in some problems that might be of use or interest to you.

(Refer Slide Time: 07:30)

The slide has a dark blue header bar with the text "Data science for Engineers". The main title "Outcomes Expectation" is centered above a list of bullet points. At the bottom right of the slide, there is a small number "4".

- **Describe** data analysis problems in a structured framework
- **Identify** comprehensive solution strategies for data analysis problems
- **Classify** and **recognize** different types of data analysis problems
- **Determine** appropriate techniques to use for the data analysis problem at hand
- **Correlate** the results of the proposed approach to the assumptions made to solve the problem
- **Judge** the appropriateness of the proposed solution based on the observed results
- **Generate** comprehensive reports for the solution framework that is adopted to solve the data analysis problem

So, in terms of an idea of what outcomes we would expect when a participant finishes this course there are many things that you can do, but these are some categories of skills that we would expect you to generate. So, you would expect you to be able to describe data analysis problems in a structured framework, once you describe that would expect you to identify some comprehensive solution strategies for the data analysis problems, classify and recognize different types of data analysis problems and at least to some level determine appropriate techniques.

Now, since we do not teach you wide variety of techniques, within the gamut of techniques that you are taught you will be able to identify an appropriate technique that you can use and in this course, we emphasize this important idea of assumption validation. So, you make some assumption support the data that you are dealing with and then those assumptions tell you what algorithms you should use and then once you run the algorithm you get the results and see whether your assumptions are validated and so

on. So, you would be able to think about how you can correlate the results of whatever you have done to the assumptions you made to solve the problem and then see whether that makes sense whether the solution makes sense and so on.

So, that is where we talk about judging the appropriateness of the proposed solution based on the observed results and ultimately, we would expect you to be able to generate comprehensive reports on the problems that you solve and then be able to say why you did, what you did, so, that is an important aspect of what we are trying to cover.

So, if you stick with us and get through all the eight weeks of this course and also diligently work on the assignments that are provided at the end of every week then we hope that you learn the fundamentals of data science, you get some fundamental grounding on important ideas and the math that you need to learn to understand data science and take this learning forward in terms of more complicated algorithms and more complicated data science problems that you might want to solve in the future.

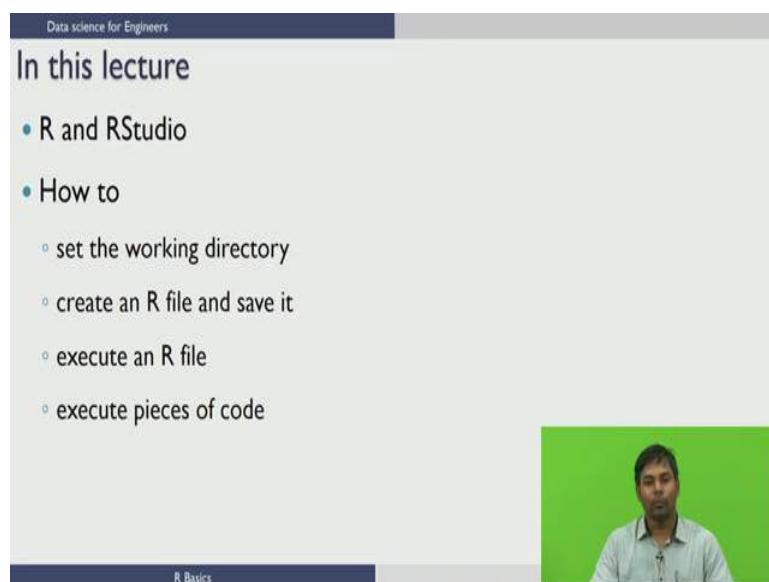
So, I hope all of you learn and enjoy from this course and we will see you as the course progresses.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 02
Introduction to R

Welcome to the course on data science for engineers. In this module, we are going to introduce R as a programming language to perform data analysis. This lecture, we are going to give a brief introduction about R and Studio.

(Refer Slide Time: 00:25)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data Science for Engineers'. Below the header, the title 'In this lecture' is displayed in bold black font. A bulleted list follows, detailing the topics covered in the lecture:

- R and RStudio
- How to
 - set the working directory
 - create an R file and save it
 - execute an R file
 - execute pieces of code

At the bottom of the slide, there is a dark blue footer bar with the text 'R Basics'. To the right of the slide content, there is a small video player window showing a person from the chest up against a green background.

In R studio, we are going to look how to set the working directory, how to create an R file and save it, how to execute an R file and how to execute pieces of R code.

(Refer Slide Time: 00:41)

Data science for Engineers

R

- Open source programming language
- Statistical software and Data analysis tool
- Interface
 - Command line user interface
- Platforms
 - Windows, Linux and macOS

- Open source programming language
- Statistical software and Data analysis tool
- Interface
 - Command line user interface
- Platforms
 - Windows, Linux and macOS

R Basics

NPTEL NOC18-CS28

Let us first see what is R. R is an open source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command line interface. R is available across widely used platforms, windows, line x and macOS Now, let us see, what is R Studio.

(Refer Slide Time: 01:05)

Data science for Engineers

RStudio

- Integrated Development Environment (IDE) for R
- Availability
 - Open source
 - Commercial
- Editions
 - Desktop
 - Server
- Platforms
 - Windows, Linux and macOS

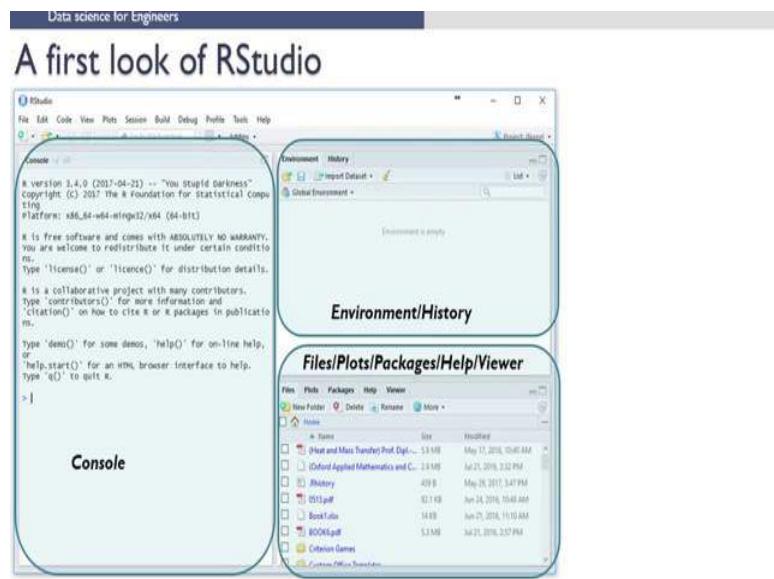
R Basics

NPTEL NOC18-CS28

R Studio is an integrated development environment for R. Integrated development environment, is a GUI, where you can write your quotes, see the results and also see the variables that are generated during the course of programming. R Studio is available as

both Open source and Commercial software. R Studio is also available as both Desktop version and Server version. For this course, we are going to use Open Source Desktop Edition so, that you can solve your assignments using this R Studio. R Studio is also available for various platforms, such as windows, line x and macOS.

(Refer Slide Time: 01:50)

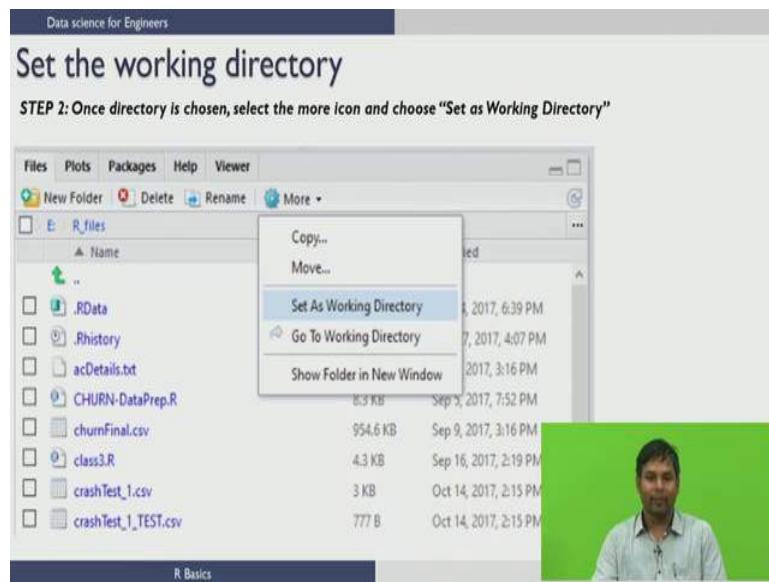


Now, let us see, how an R Studio looks, when you first run application. This is how an R Studio Interface looks. When you first run the application, to the left, we see Console panel, where you can type in the comments and see the results that are generated when you type in the commands. To the top right, you have Environmental History pane. It contains 2 types: the Environment type, where, it shows the variables that are generated during the course of programming, in a workspace, which is temporary and in the History tab, you will see all the commands that are used till now from the beginning of usage of R Studio. The right bottom, you have another panel, which contains multiple tab, such as files, plots, packages and help.

The Files tab shows the files and directories that are available in the default workspace of R. The Plots tab shows the plots that are generated during the course of programming. And the Packages tab helps you to look, what are the packages that are already installed in the R Studio and it also gives an user interface, to install new packages. The Help tab is a most important one, where you can get help from the R Documentation on the functions that are in built in R. The final and last tab is the Viewer tab, which can be

used to see the local web content that is generated using R, are some other application. For this course, you are not going to use this tab from much. So, we limit ourself not discuss more about that, viewer tab. So, we have got an idea about how R Studio looks. Let us see, how to set the working directory in R Studio.

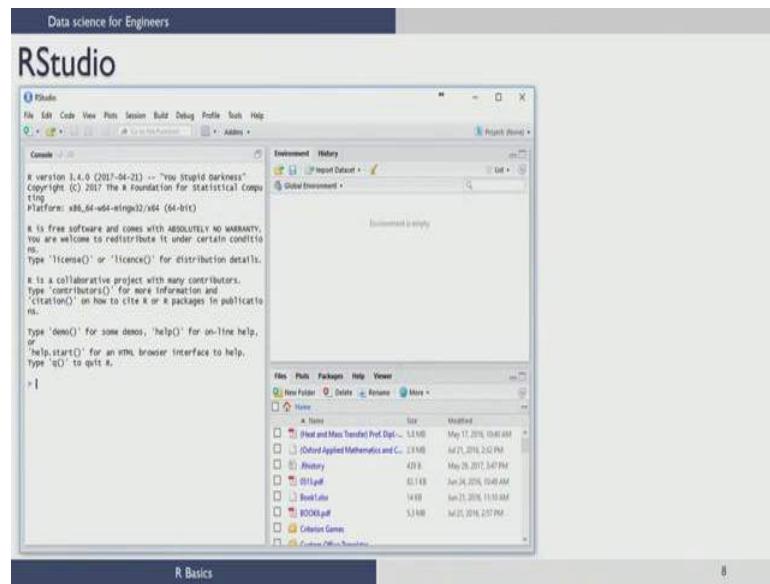
(Refer Slide Time: 03:52)



The working directory in R Studio can be set in 2 ways. The first, way is to use the console and using the command Set working directory. You can use this function Set working directory and give the path of the directory which u want to be the working directory for r studio, in the double codes.

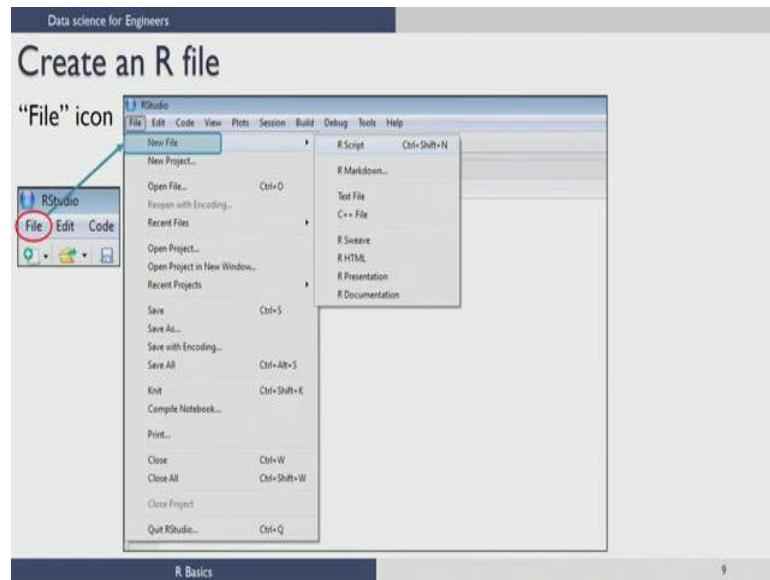
R, to set the working directory from the GUI, you need to click on this 3 dots button. When you click this, this will open up a file browser, which will help you to choose your working directory. Once you choose your working directory, you need to use this setting button in the more tab and click it and then you get a popup menu, where you need to select Set as working directory. This will select the current directory, which you have chosen using this file browser as your working directory.

(Refer Slide Time: 04:50)



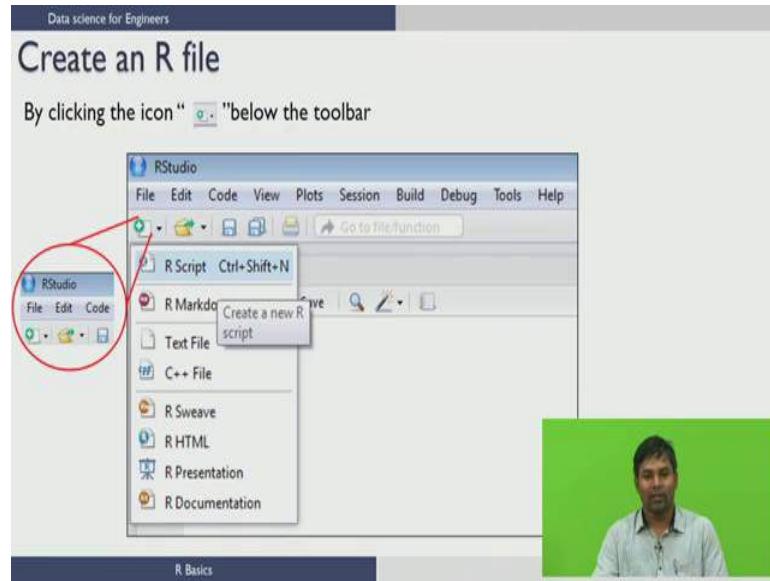
Once you set the working directory, you are ready to program in R Studio.

(Refer Slide Time: 04:56).



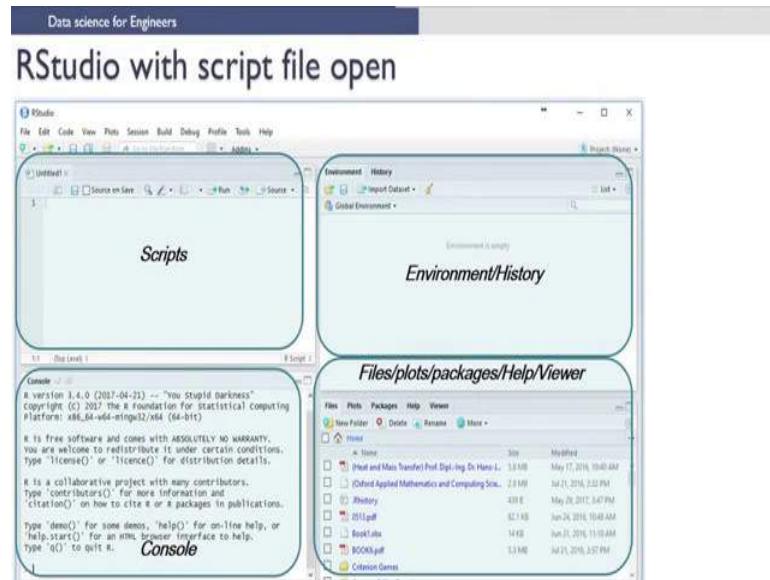
Let us illustrate how to create an R file and write some code. To create an R file, there are 2 ways: The first way is: you can click on the file tab, from there when you click it will give a drop down menu, where you can select new file and then R script, so that, you will get a new file open.

(Refer Slide Time: 05:18)



The other way is to use the + button, that is just below the file tab and you can choose R script, from there, to open a new R script file.

(Refer Slide Time: 05:30)



Once you open an R script file, this is how an R Studio with the script file open looks like. So, 3 panels console environmental history and files and plots panels are there. On top of that, you have a new window, which is now being opened as a script file. Now you are ready to write a script file or some program in R Studio.

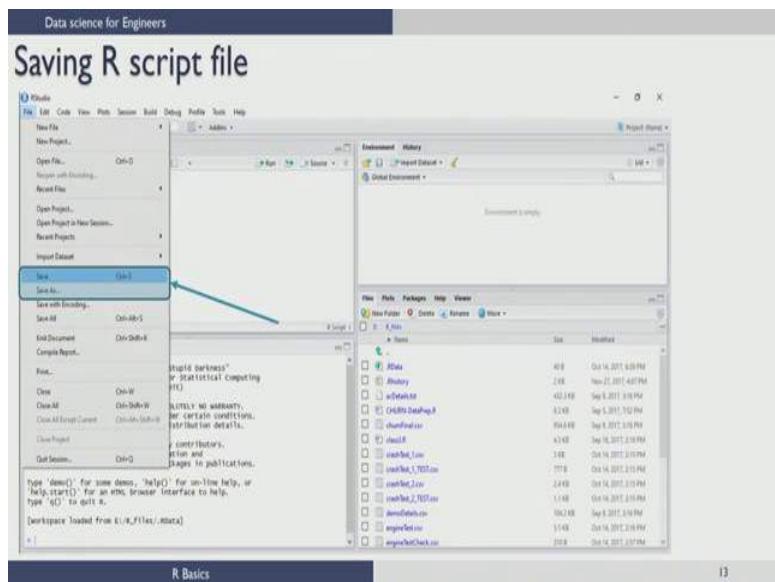
(Refer Slide Time: 05:51)

```
1 a = 11
2 b = a*10
3 print(c(a,b))
```

The screenshot shows the RStudio interface. The top bar has tabs for 'Data science for Engineers' and 'Writing scripts'. The main area shows an 'Untitled.R' script with the above code. Below it is the 'Console' tab showing R version information and a warning about redistribution. To the right is the 'Environment' pane showing the 'Global Environment' with no objects listed. At the bottom is the 'R Basics' pane.

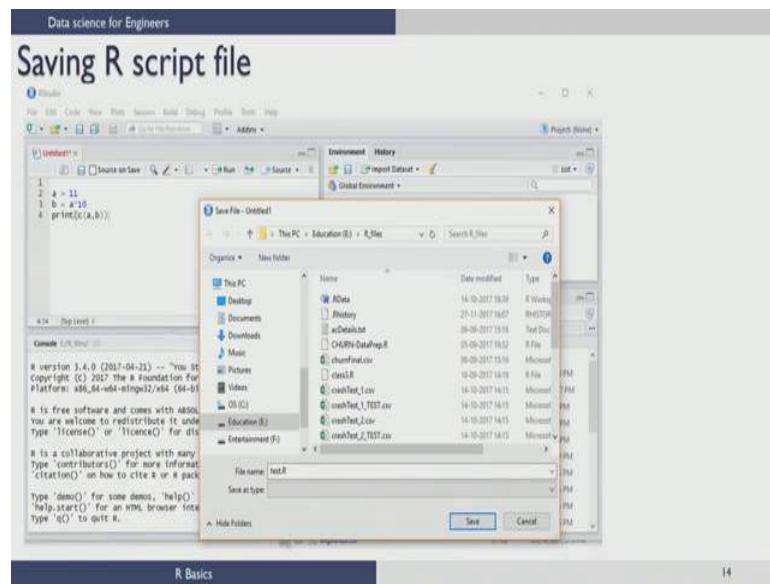
So, let us illustrate this with a small example, where I am assigning a value of 11 to a, in the first line of the code which I have written and you have b which is a times 10, that is the second command, I am evaluating the value of a times 10 and assign the value to the b and the third statement, which is print c of a, b concatenates this a and b and print the result. So, this is how you write a script file in R. Once you write a script file, you have to save this file before you execute it.

(Refer Slide Time: 06:37)



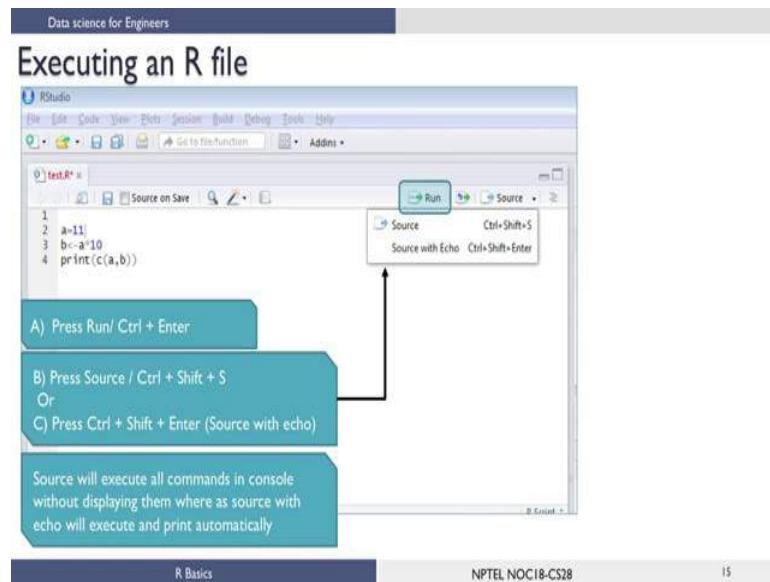
Let us see, how to save the R file. From the file menu, if you click the file tab, you can either save the file, when you want to save the file, if you click the save button, it will automatically save the file has untitled x. So, this x can be 1 or 2 depending upon how many R scripts you have already opened, or it is a nice idea, to use the Save as button, just below the Save one, so that, you can rename the script file according to your wish. Let us suppose we have click the, Save as button.

(Refer Slide Time: 07:14)



This will pop out a window like this, where you can rename the script file as test R, are the one which you are intended to. Once you rename, you can say save, that will save the script file.

(Refer Slide Time: 07:31).



So now, we have seen how to open an R script and how to write some code in the R script file. The next task is to execute the R file. There are several ways you can execute the commands that are available in the R file. The first way is to use run command.

This run command, can be executed using the GUI, by pressing the run button there, or you can use the Shortcut key, this is control + enter, what it does is, it will execute the line in which the cursor is there. The other way is to run the R code 'R' using source R source with echo. The difference between source and source with echo is the following: The Source command executes the whole R file and only prints the output, which you wanted to print. Whereas, source with echo prints the commands also, along with the output you are printing.

(Refer Slide Time: 08:38)

```
1 a = 11
2 b = a*10
3 print(c(a,b))

[1] 11 110
```

So, this is an example, where I have executed the R file, using the source with echo, you can see, in the console, that it printed a the command a = 11 and the command b = a time 10 and also the output print c of a, b with the values. So, a = 11 and b = 11 times 10, this is 110. So, this is how, the output will be printed in console. So, that is the result.

(Refer Slide Time: 09:14)

```
1 a = 15
2 b = a*10
3 print(c(a,b))

[1] 15 150
```

Now, let us see how to execute the pieces of code in R. As you have seen earlier, you can use run command, to run the single line, right. So now, let us try to assign value 14 for a and then try to run it. So, how do you do this? Take your cursor to the line, which you

want to edit, replace that 11 by 14 and then use control enter or the run button. This will execute only the line, where the cursor is placed.

(Refer Slide Time: 09:42)

The screenshot shows the RStudio interface. In the Environment pane, there are two entries: 'a' with a value of 14 and 'b' with a value of 110. A message box in the center of the screen displays the text 'Value of 'a' has changed but not 'b''. Below the message box is the standard RStudio menu bar with options like Files, Plots, Packages, Help, and Viewer. Under the 'Files' menu, there is a list of files in a folder named 'R_files'. The files listed are '.RData' (408 KB, Oct 14, 2017), '.Rhistory' (2 KB, Nov 27, 2017), and 'acDetails.txt' (432.3 KB, Sep 9, 2017). In the bottom right corner of the interface, there is a small video window showing a person speaking.

In the Environment pane, you can see that, only value of a, has been changed and the b value remains same. This is because, we have executed only the line 2 of the code, which change the value of a, but we have not executed the code of line 3. So, the b value remains as is. Value of a, has changed, but not the value of b.

(Refer Slide Time: 10:05)

The screenshot shows the RStudio interface with a title 'Executing R files – Summary'. Below the title is a bulleted list of points:

- Run can be used to execute selected lines
- Source/ Source with echo is for a whole file
- Advantages – using Run :
 - troubleshooting/debugging
- Disadvantages – using Run :
 - For large section, console will be over populated and messy

In the bottom right corner of the interface, there is a small video window showing a person speaking.

In summary, we can say that, Run can be used to execute the selected lines of R code. Source and Source with echo can be used to run the whole file. The advantage of using Run is, you can troubleshoot or debug the program when something is not behaving according to your expectations. The disadvantages of using run command is, it populates the console and make it messy unnecessarily.

In the next lecture, we are going to see how to add comments to the R file and how to add comments to the single line and multiple lines etc.

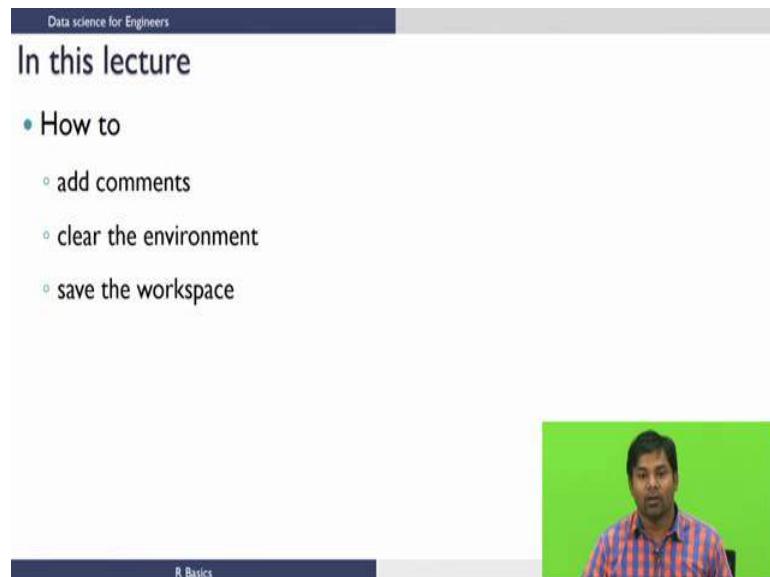
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 03
Introduction to R

Welcome to the lecture 2 in the R model of the course Data science for Engineers. In the previous lecture we have given a brief introduction about R and R studio and we have seen how to create an R file write some codes in R file and how to execute an R file.

(Refer Slide Time: 00:34)



Data science for Engineers

In this lecture

- How to
 - add comments
 - clear the environment
 - save the workspace

R Basics



In this lecture we are going to show how to add comments to the R file, how to clear the environment and how to save the workspace of R now let us first look at how to add comments to the R file.

(Refer Slide Time: 00:51)

The slide has a dark blue header bar with the text 'Data science for Engineers' in white. Below the header, the title 'Why to add comments?' is displayed in a large, bold, black font. A bulleted list follows, explaining the reasons for adding comments:

- Improve the readability of code
 - the purpose of the code
 - explain algorithms used to accomplish the purpose
- To generate documentation external to the source code itself by documentation generators

At the bottom left of the slide, there is a small dark blue bar with the text 'R Basics' in white. On the right side of the slide, there is a video frame showing a man with dark hair and a beard, wearing a red and blue plaid shirt, sitting in front of a green screen.

Before that let us ask this question: why do you add comments to your codes? Adding comments improve the readability of your code for example, you can explain the purpose of the code you are writing in the comments or you can explain what an algorithm is doing to accomplish the purpose which you are attempting at. Writing comments also help us to generate documentation which is external to the source code itself by documentation generators.

(Refer Slide Time: 01:29)

The slide has a dark blue header bar with the text 'Data science for Engineers' in white. Below the header, the title 'Add comments –single line' is displayed in a large, bold, black font. A text block provides instructions on how to comment a single line:

To comment a single line, insert '#' at the start of the
comment

Below the text, there is a screenshot of an R script editor window titled 'comments.R'. The code in the editor is as follows:

```
1 # Author: Tarnuru Hemanth Kumar
2 # This program takes a single number and calculates a
3 # value of 10 times of it
4 a = 10 # the input number
5 b = 10*a # b is calculated as 10 times "a"
6 # "=" and "<-" both can be used as the assignment operators
7 # the next statement prints the output
8 print(c(a,b))
```

At the bottom left of the slide, there is a small dark blue bar with the text 'R Basics' in white. At the bottom right, there is a small number '4'.

Let us look how to add comments to a single line in R script first you can comment a single line R by using hash key at the start of the comment if you see in this example I have commented this first comment by a hash key which turns this command green and if you notice these commands are describing what this program is doing, what it is doing is it is taking a single number and then calculating a value which is 10 times of it.

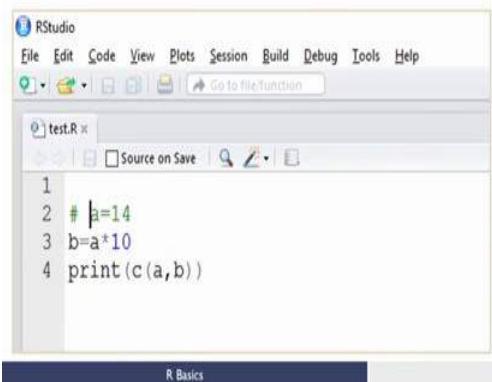
So, you can see here I am defining a variable $a = 10$ which I am commenting it out as the input number and now I am explaining this operation which is being happened here which is b is calculated as 10 times a and if you would have remembered in the previous lecture we have used this symbol for assigning a value to a variable you can also use $=$ in R studio that is been demonstrated here. Now you can see how commenting makes your script file more readable.

(Refer Slide Time: 02:41)

Data science for Engineers

Add comments –single line

To make a line of code inert, insert '#' at the start of the line



The screenshot shows the RStudio interface with a code editor window titled "test.R". The code contains the following lines:

```
1 # a=14
2 b=a*10
3 print(c(a,b))
```

A small video player window in the bottom right corner shows a man speaking.

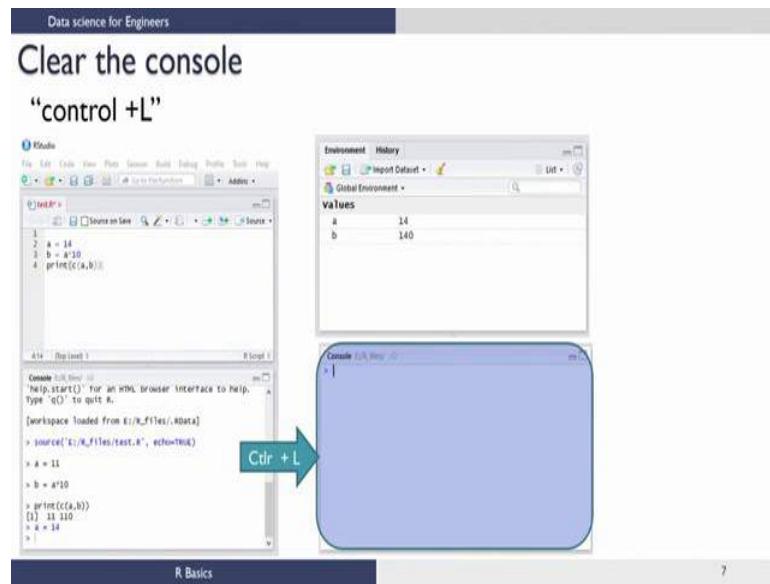
Comments can also be used to make certain lines of code inert you can do that by inserting a hash key at the beginning of the line like here you can see I want to comment this line which says $a = 14$ if I wish to do so, I can comment it by keeping a hash key in front of it. Now we will see how to add comments to multiple lines at once in R.

(Refer Slide Time: 03:10)

There are 2 ways first we used to select the multiple lines which you want to comment using the cursor and then use the key combination control + shift + C to comment or uncomment the selected lines.

the other way is to use the GUI, select the lines which you want to comment by using cursor and in the code menu if you click on the code menu a pop up window pops out in which we need to select comment or uncomment lines which appropriately comments or uncomment the lines which you have selected. In some cases when you run the codes using source and source with echo your console will become messy.

(Refer Slide Time: 03:57)



And it is needed to clear the console let us now look at how to clear the console. The console can be cleared using the shortcut key control + L, let us look at an example, in this code I have defined a and calculated b and printed a comma b, when I execute this code using source with echo all the commands will get printed here. Now, let us say suppose I want to clear this console what I have to do is I have to click here and I have to enter the key combination control + L. Once I do this you can see that the console will get cleared remember clearing console will not delete the variables that are there in the workspace you can see that even though we have cleared the console in the workspace we still have the variables that are created earlier.

(Refer Slide Time: 04:51)

Data science for Engineers

Clear the environment -rm()

Single variable: Enter in console/R script : rm(variable)

All variables: Enter in console/R script : rm(list=ls())

OR

Environment History

Import Dataset

Global Environment

values

a	14
b	110

R Basics

A video thumbnail of a man speaking is visible in the bottom right corner.

Now, let us see how to clear the variables from the R environment you can clear the variables on the R environment using rm command, when you want to clear a single variable from the R environment you can use the rm function has shown here rm followed by the variable you want to remove. If you want to delete all the variables that are there in the environment what you can do is you can use the rm with an argument list = ls followed by parenthesis or you can clear all the variables in the environment using the GUI in the environment history pane you see this brush button, when you press the brush button it will pop up

(Refer Slide Time: 05:38)

Data science for Engineers

Confirmation dialog

Are you sure you want to remove all objects from the environment? This operation cannot be undone.

Yes No

Console

```
a <- 14  
b <- a*10  
print(c(a,b))  
[1] 14 110  
source('E:/R_files/test.R', echo=TRUE)  
> a = 11  
> b = a*10  
> print(c(a,b))  
[1] 11 110  
> a = 14
```

Environment History

values

a	14
b	110

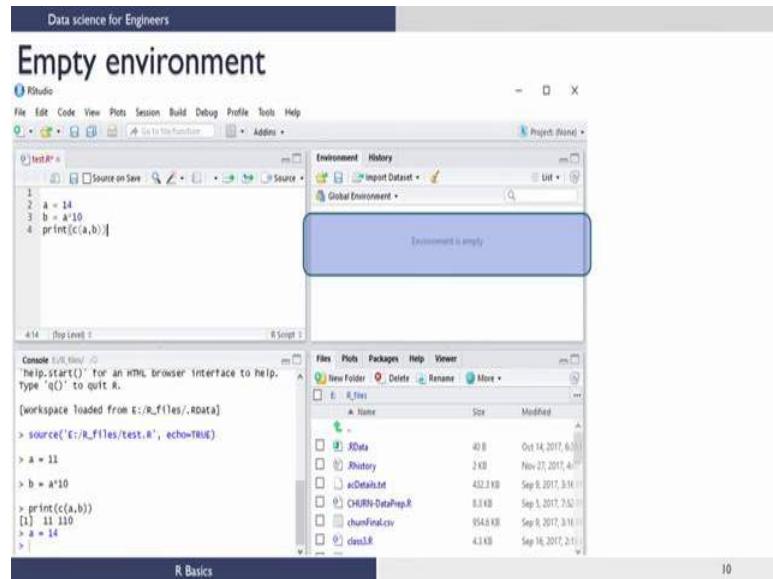
Project Blanks

R Basics

A video thumbnail of a man speaking is visible in the bottom right corner.

a window saying we want to clear all the objects that are available in environment if you say yes it will clear all the variables.

(Refer Slide Time: 05:48)



Which is shown there and you can see the environment is empty now. Now, let us see how to save.

(Refer Slide Time: 05:55)

A screenshot of the RStudio interface. The title bar says "Data science for Engineers". The main window has a dark header with "Saving data from workspace" and a light blue footer with "R Basics".

Workspace data

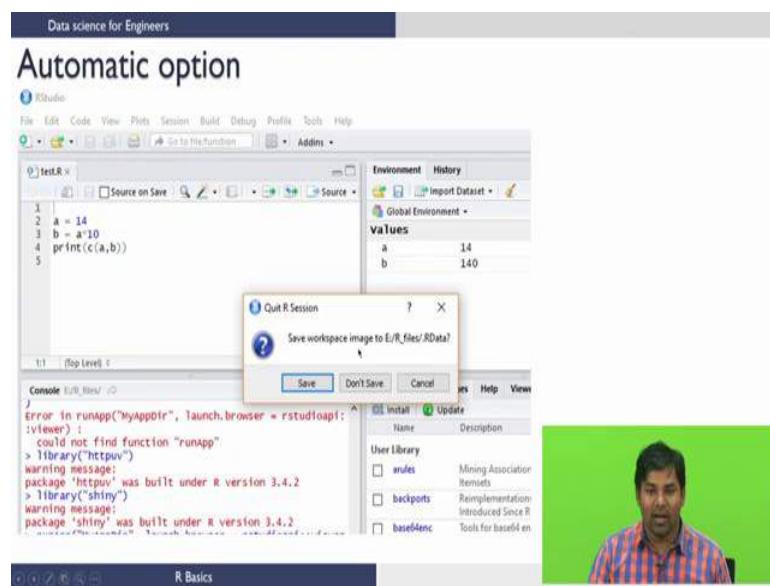
- Workspace information is temporary
- Is not retained after the session
 - If you close the R-session
 - If you restart RStudio

The data from the workspace in R I have already mentioned that the information that is saved in the environment of R is temporary and it is not retain when you close the R

session or restart the R Studio it is sometimes needed to save the data which is already there in the current session.

The reason being you would have done certain operations to get the data to this form and you do not want to repeat those actions and you need to start from the point where you want to leave now, in that cases you need to save the data from the R environment when you want to do that.

(Refer Slide Time: 06:36)



There are 2 ways the first one is the automatic option when you close the R Studio application it will ask you look do you want to save the workspace image if you say yes it will save all the variables that are there in the workspace, if you say do not save the R Studio will exit and the workspace information not be saved.

(Refer Slide Time: 06:57)

The screenshot shows a presentation slide titled "Manual saving". The slide content includes a bulleted list of two items, a code example, and a video player interface.

- Can be permanently saved in a file – save command
- Can be reloaded for future sessions – load command

```
# Example code
save(a,file="sess1.Rdata") # to save a single variable 'a'
# to save a full workspace with specified file name
save(list=ls(all.names=TRUE),file="sess1.Rdata")
save.image() # short cut function to save whole workspace
load(file="sess1.Rdata") # to load saved workspace
```

R Basics

A video player window is visible on the right side of the slide, showing a man from the chest up, wearing a striped shirt, speaking. The background of the video player is green.

You can also save the workspace information using manual method where you can save the information to a file using the save command and the saved information can be reloaded for the future sessions using the load command let us see how to do that in R. Here is an example code the first line here shows how to save a variable that is there in the workspace into a file name sess1 dot R data.

So, in the comments you can see that this is the command which you can use to save a single variable a, if you are willing to say the full workspace you need to use this command save list = ls with argument all dot names = true and you can give the filename whatever you wish to and the shortcut key for this command which is given here is save dot image which saves the data in the environment into dot R data file in the current working directory. Once you do that you can load the workspace information at later point of time whenever you want using this command load you can specify the file = the file name which you save the data into.

So, in this lecture we have seen how to add comments to R file, how to clear the console and how to clear the R objects that are there in the environment and also we have seen how to save the variables that are available in the R environment for further use. In the next lecture we are going to introduce you to the basic data types of R.

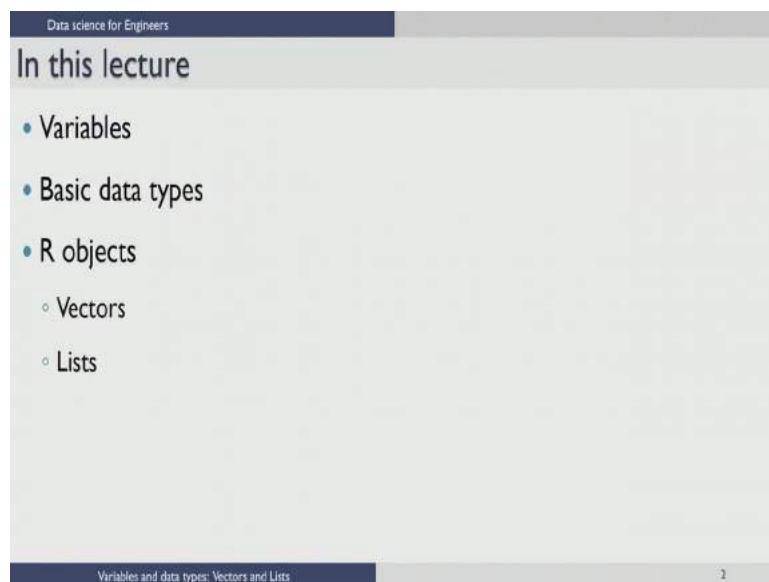
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 04
Variables and datatypes in R

Welcome, to the lecture – 3 of R module in the course Data Science for Engineers.

(Refer Slide Time: 00:21)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header, the title 'In this lecture' is displayed in a large, bold, dark blue font. A bulleted list of topics follows, also in a dark blue font. The topics are:

- Variables
- Basic data types
- R objects
 - Vectors
 - Lists

At the bottom of the slide, there is a footer bar with the text 'Variables and data types: Vectors and Lists' on the left and the number '2' on the right.

In this lecture we are going to see the rules for naming the variables in R and what are the basic data types that are available in R and we are also going to see two basic R objects; vectors and lists, in detail.

(Refer Slide Time: 00:36)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header is a large white area with rounded corners containing the word 'Variables'. In the bottom right corner of this white area, there is a small video frame showing a man from the chest up, wearing a light blue shirt, standing in front of a green screen. At the very bottom of the slide, there is a thin dark blue footer bar with the text 'Variables and data types: Vectors and Lists'.

Let us see the rules for naming the variables in R first.

(Refer Slide Time: 00:41)

This screenshot shows a more detailed slide on variable naming. The top part is identical to the previous slide, with 'Data science for Engineers' in the header and 'Variables' in the main title area. The main content is divided into two columns. The left column, titled 'Variable names - Rules', contains a bulleted list of three items: 'Allowed characters are Alphanumeric, '_' and '.'', 'Always start with alphabets', and 'No special characters like !,@,#,\$,...'. The right column, titled 'Examples', is split into two sections: 'Correct naming:' which lists examples like > b2 =7 and > Manoj_GDPL = "Scientist"; and 'Wrong naming:' which lists examples like > 2b =7 and 'Error: unexpected input in "2b"'.

The variable name in R has to be alphanumeric characters with an exception of underscore and period, the special characters which can be used in the variable names. The variable name has to be started always with an alphabet and no other special characters except the underscore and period are allowed in the variable names. This shows some examples of the correct variable names that can be used in R. The first one,

`b2 = 7`, assigns the value of 7 to the variable `b2`. This is a valid variable name because it started with an alphabet and it has only alphanumeric characters.

Similarly, the second variable `Manoj_GDPL = scientist` this is also valid variable name because it has a special character, but it is underscore which is allowed special character for the variable names. Now, let us see some examples where the variable names are not correct the variable `2b = 7`, gives an error because that variable name has started with the numeric character which is not following the rules for the names of the variables in R.

(Refer Slide Time: 02:00)

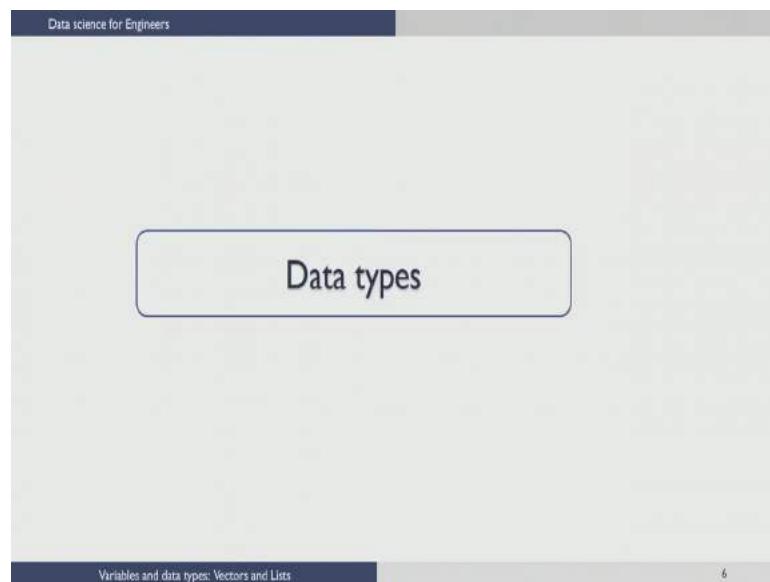
The screenshot shows the RStudio interface with the following details:

- Data science for Engineers** is displayed at the top left.
- Predefined constants** is the active tab in the sidebar.
- Constant** and **Symbol in R** are the columns of the table.
- Constant** values:
 - Pi**: pi
 - letters**: a,b,c,.....x,y,z
 - LETTERS**: A,B,.....X,Y,Z
 - Months in a year**: month.name, month.abb
- Symbol in R** values:
 - Pi**: pi
 - letters**: letters
 - LETTERS**: LETTERS
 - Months in a year**: month.name, month.abb
- Console** output window shows the following R session:


```
> pi
[1] 3.141593
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i"
[10] "j" "k" "l" "m" "n" "o" "p" "q" "r"
[19] "s" "t" "u" "v" "w" "x" "y" "z"
> LETTERS
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I"
[10] "J" "K" "L" "M" "N" "O" "P" "Q" "R"
[19] "S" "T" "U" "V" "W" "X" "Y" "Z"
> month.name
[1] "January" "February" "March"
[4] "April" "May" "June"
[7] "July" "August" "September"
[10] "October" "November" "December"
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun"
[7] "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```
- Variables and data types: Vectors and Lists** is the footer text.
- 5** is the page number in the bottom right corner.

R also contains some predefined constants that are available such as `pi`, `letters`, the lowercase `a` to `z` and `letters` in the uppercase which are uppercase letters from `A` to `Z` and `months in a year`, you can have full month name by `month.name` and you can have abbreviated month names by typing `month.abb`.

(Refer Slide Time: 02:25)



Let us now look at the data types that are available in the R.

(Refer Slide Time: 02:31)

The slide is titled "Basic data types". It features a table with two columns: "Basic data types" and "Values". The table rows are: Logical (TRUE and FALSE), Integer (Set of all integers, Z), Numeric (Set of all real numbers), Complex (Set of complex numbers), and Character ("a", "b", "c", ..., "X", "y", "z", "@", "#", "\$", etc., "1", "2", etc.). To the right of the table is a video player showing a man speaking against a green background.

Basic data types	Values
Logical	TRUE and FALSE
Integer	Set of all integers, Z
Numeric	Set of all real numbers
Complex	Set of complex numbers
Character	"a", "b", "c", ..., "X", "y", "z", "@", "#", "\$", etc., "1", "2", etc..

Variables and data types: Vectors and Lists

R has the following basic data types and this table shows the data type and the values that each data type can take. So, R has logical data types which take either a value of true or false, it supports integer data types which is the set of all integers and numeric which is set of all real numbers. We can also define complex variables, R supports set of all the complex numbers. Also, we can have a character data type where you have all the

alphabets and special characters which are under the window of basic data types of characters. There are several task that can be done using data types.

(Refer Slide Time: 03:14)

TASK	ACTION	SYNTAX/EXAMPLE
Find data type of object	use command "typeof()"	Syntax: <code>typeof(object)</code>
Verify if object is of a certain datatype	use prefix "is." before datatype as command.	Syntax: <code>is.data_type(object)</code> Example : <code>is.integer()</code>
Coerce or convert data type of object to another	use prefix "as." before datatype as command.	Syntax: <code>as.data_type(object)</code> Example : <code>as.logical()</code>

Note : Not all coercions are possible and if attempted will return "NA" as output

Sample Codes

```

Console > / 
> typeof(1)
[1] "double"
> typeof("22-01-2001")
[1] "character"

Console > / 
> is.character("21-11-2001")
[1] TRUE
> is.character(as.Date("21-11-2001"))
[1] FALSE

Console > / 
> as.complex(2)
[1] 2+0i
> as.numeric("a")
[1] NA
  
```

Variables and data types: Vectors and Lists

The following table gives you the task action and the syntax for doing the task. For example, the first task is to find the data type of the object. To find the data type object you have to use type of function. The syntax for doing that is you need to pass the object as an argument to the function type of to find the data type of an object

The second task is, to verify if object is of certain data type. To do that you need to prefix is dot before the data type as a command. The syntax for that is, is dot, data type of the object you have to verify. For example, if you have variable a, which is defined as an integer and if you use this command is dot integer of a, it will show true originally.

The variable a is not defined as integer this will show false and the third task is interesting task where you can change the or convert the data type of one object to another to perform this action you have to use, as dot, before the data type as the command; the syntax for doing that is as dot data type of the object which you want to coerce. Note that all the coercions are not possible and it attempted will be returning a null value.

There are sample codes for doing this as which are in the bottom. The first one is type of 1, so, 1 is a numeric variable. So, if you say, type of, you will get double which is

numerical variable and if you say, type of a string, that is printed 22-1-2001, it will give value as character. Now, if you going to ask whether this; the character variable which have created 21-11-2001 is this character type variable, you can use this command is dot character the result is true because you have defined it as a character variable.

The next example, here what you are going to do here is we are coercing the character variable which is defined earlier that is 22-11-2001 as date and then you are checking whether that date is a character variable. The result is false because when you coerce this character variable as a date it will be a numeric variable, when you want to ask whether this variable is a character the result will be false.

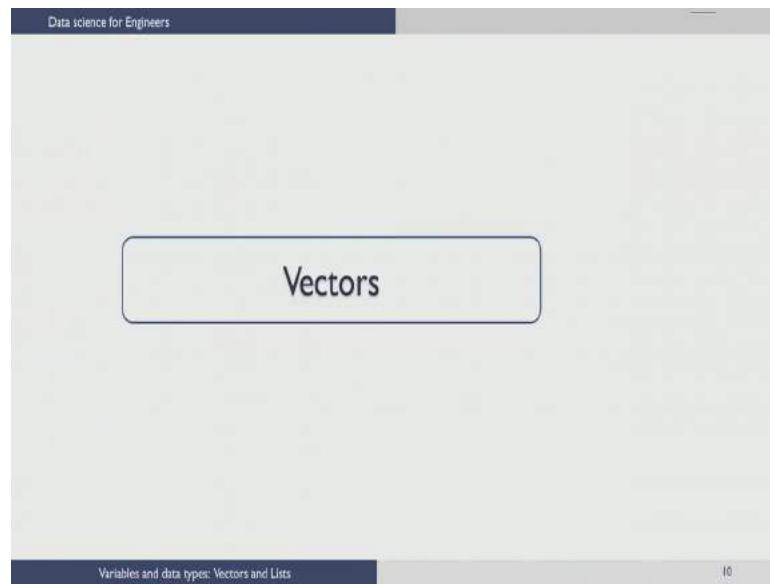
You can also coerce numeric variable into complex variable by using as dot complex of, so, for example, we have as dot complex of 2, will convert this numeric variable 2 into the complex variable $2 + 0i$. Now, let us try coercing a character into a numeric variable using this command as dot numeric of which has given us not available or NA. This means the coercion from the characters to numeric numerical variables is not possible.

(Refer Slide Time: 06:48)

Object	Values
Vector	Ordered collection of same data types
List	Ordered collection of objects
Data frame	Generic tabular object

We have several basic objects of R, in this the most important ones are; vectors, lists and data frames. A vector is an ordered collection of same data types, list is ordered collection of object themselves and data frame is a generic tabular object which is very important and the most widely used objects of R programming language. We will see in detail about each of this in the coming parts of the lecture and the other lectures also.

(Refer Slide Time: 07:22)



Let us now first see, what is a vector?

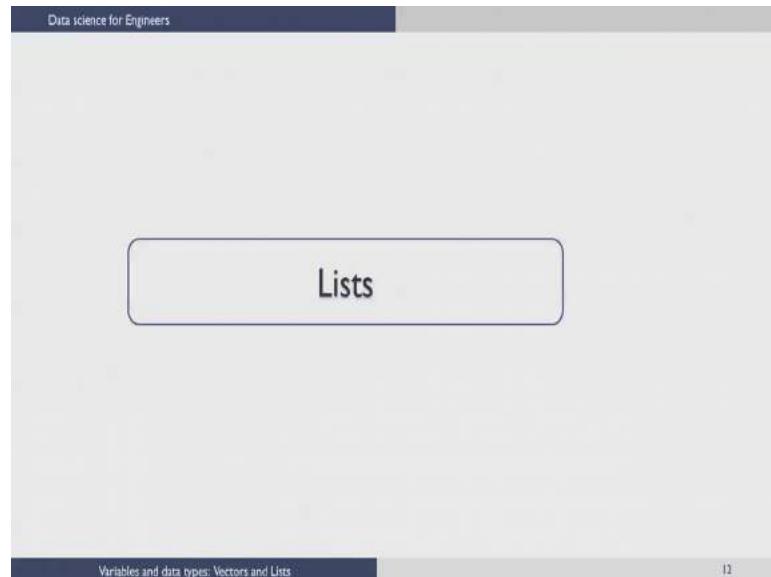
(Refer Slide Time: 07:25)

A screenshot of a presentation slide titled "Vectors". The slide has a dark blue header bar with the text "Data science for Engineers" and a light blue footer bar with the text "Variables and data types: Vectors and Lists". The main content area contains a list of bullet points: "Vector : an ordered collection of basic data types of given length" and "All the elements of a vector must be of same data type". Below this, there is a "Code" section and a "Console Output" section. The code section shows R code: "# Vectors Example", "X = c(2.3,4.5,6.7,8.9)", and "print(X)". The console output section shows the R session: "Console > # Vectors Example", "Console > X = c(2.3,4.5,6.7,8.9)", "Console > print(X)", "[1] 2.3 4.5 6.7 8.9", "Console >".

Vector is an ordered collection of basic data types of a given length. So, only key thing here is all the elements of a vector must be of a same data type. If you want to see an example the way you creating vector in R is using the concatenation command that is C. So, now I am going to define a vector which is containing four numeric variables and I am assigning it to a variable X. This is what the code here X = concatenation of these numbers and then I am printing X. So, if you execute this piece of code, this is how the

output in the console looks. It creates a vector X with the variables 2.3, 4.5, 6.7, 8.9 and prints them in the console.

(Refer Slide Time: 08:21)



Next, we move onto list.

(Refer Slide Time: 08:23)

A screenshot of a presentation slide titled "Data science for Engineers". The main title is "Lists in R: create a list". Below the title, there is a bulleted list:

- List : a generic object consisting of an ordered collection of objects
- A list could consist of a numeric vector, a logical value, a matrix, a complex vector, a character array, a function, and so on

Below the list, there are two sections: "Code" and "Console Output".

Code

```
# List Example : Employee details
ID = c(1,2,3,4)
emp.name =c("Man","Rag","Sha","Din")
num.emp = 4
emp.list = list(ID, emp.name,num.emp)
print(emp.list)
```

Console Output

```
> ID = c(1,2,3,4)
> emp.name =c("Man","Rag","Sha","Din")
>
> num.emp = 4
> emp.list = list(ID, emp.name,num.emp)
> print(emp.list)
[[1]]
[1] 1 2 3 4
[[2]]
[1] "Man" "Rag" "Sha" "Din"
[[3]]
[1] 4
```

List is a generic object consisting of ordered collection of objects. List can be a list of vectors, list of matrices, list of characters and list of functions and so on. To illustrate how a list looks, we take an example here. We want to build a list of employees with the

details for this I want the attributes such as ID, employee name and number of employees. So, I am creating each vector for those attributes.

The first attributes is a numeric variable containing the employee IDs which is created using the command here, which is a numeric vector and the second attribute is employee name which is created using this line of code here, which is the character vector and the third attribute is number of employees which is a single numeric variable.

Now, I can combine all these three different data types into a list containing the details of employees which can be done using a list command. So, this command here creates m dot list variable which is a list of the ID, emp dot name and num dot employees that are defined above.

Once you create a list you can print the list and see how the output looks. So, when you execute this course, you can see in the console the list is printed this is the first one IDs 1, 2, 3, 4; this is the second element of the list which are contain the names of employees and the third element of the list which are saying how many number of employees are available. So, we have created a list.

Now, we can see how to access the components of the list.

(Refer Slide Time: 10:12)

The screenshot shows a slide titled "Accessing components (by names)" from a Data Science for Engineers course. The slide contains two main sections: "Code" and "Console Output".

Code:

```
# Continue after first 4 lines of R
# code from previous example
emp.list = list("Id"=ID,
                "Names" = emp.name,
                "Total staff"=num.emp)
print(emp.list$Names)
```

Console Output:

```
> # Continue after first 4 lines
> # of R code from previous example
> emp.list = list("Id"=ID,
+                 "Names" = emp.name,
+                 "Total staff"=num.emp)
> print(emp.list$Names)
[1] "Man" "Rag" "Sha" "Din"
>
>
```

All the components of a list can be named and you can use that names to access the components of the list. For example, this is the same list we have created you can use the

same ID, emp dot name and emp dot employee. Instead of directly creating a list you can also give the names for this attributes as ID, names of employees and the total staff as shown in the code here. Once you execute this code we can see now that list is created and if you want to access this element of the list you can do that by using the dollar command m dot list is the list and you want access the component with the name, names.

So, when you use this command and print the result you can see the names of the employees that are printed.

(Refer Slide Time: 11:14)

Data science for Engineers

Accessing components (indices)

To access top level components, use double slicing operator " [[]]" or [] and for lower/inner level components use "[]" along with " [[]]"

Code	Console Output
# continuing from previous # code print(emp.list[[1]]) print(emp.list[[2]]) print(emp.list[[1]][1]) print(emp.list[[2]][1])	Console > print(emp.list[1]) \$Id [1] 1 2 3 4 Console > print(emp.list[2]) \$Names [1] "Man" "Rag" "Sha" "Din" Console > print(emp.list[[1]][1]) [1] 1 Console > print(emp.list[[2]][1]) [1] "Man" >

Variables and data types: Vectors and Lists 15

You can also access the components of the list using indices. To access the top level components of a list you have to use double slicing operator which is two square brackets and if you want access the lower or inner level components of a list you have to use another square bracket along with the double slicing operator. The course here illustrates how to access the top level components; for example, I want access the IDs, I can use print emp dot list and this is a double slicing operator which will give me the first level which is ID.

The second component can also be similarly accessed that is the result is shown here and if you want access, for example, the first sub element or the inner component of the component ID you have to use emp dot list the double slicing operator and the first element in the another square bracket.

Similarly, you can access the first employee name using double slicing operator to be followed by the element one which prints the value man from the employee list.

(Refer Slide Time: 12:35)

The slide has a header 'Data science for Engineers' and a title 'Manipulating lists'. A subtitle in a box states 'A list can be modified by accessing components & replacing them'. The 'Code' pane contains:

```
# continuing from previous code
emp.list["Total staff"]=5
emp.list[[2]][5]="Nir"
emp.list[[1]][5]=5
print(emp.list)
```

The 'Console Output' pane shows the R session:

```
> # continuing from previous code
> emp.list["Total staff"]=5
> emp.list[[2]][5]="Nir"
> emp.list[[1]][5]=5
> print(emp.list)
$Id
[1] 1 2 3 4 5
$Names
[1] "Man" "Rag" "Sha" "Din" "Nir"
$`Total staff`
[1] 5
```

A video player interface is visible on the right, showing a person speaking against a green screen.

A list can also be modified by access in the components and replacing them with the ones which you want. For example, I want to change the total number of staff into 5, that can be done easily by assigning a value 5 to the total staff and I want to add a new employee name to the list the component of the list which has the employee names is 2 and I want to add this new name Nir as a new employee to that sub component.

So, I can directly assign this character variable Nir to the second component and fifth sub component of the list. Now, we need to also increase the employee ID and you have to give this employee and new ID which is 5, what we are doing now, in this command is your accessing the fifth sub element of the level one component and then assigning data value of 5.

Now, once you print the list you can see that the IDs, number of employees are 5 and total staff is 5 and the name Nirav is getting added to the list.

(Refer Slide Time: 13:50)

The screenshot shows a RStudio interface. The title bar says "Data science for Engineers". The main area has a section titled "Concatenation of lists" with the sub-instruction: "Two lists can be concatenated using the concatenation function, c(list1, list2)". Below this is a "Code" pane containing R code:

```
# continuing from previous code
emp.ages = list("ages" = c(23,48,5
4,30,32))
c(23,48,54,30,32)
emp.list= c(emp.list , emp.ages)
print(emp.list)
```

To the right is a "Console Output" pane showing the results of the code execution:

```
Console / 
> emp.ages = list("ages" = c(23,48,5
4,30,32))
> emp.list= c(emp.list , emp.ages)
> print(emp.list)
$Id
[1] 1 2 3 4 5

$Names
[1] "Man" "Rag" "Sha" "Din" "Nir"

$Total staff
[1] 5

$ages
[1] 23 48 54 30 32
```

At the bottom left is a footer: "Variables and data types: Vectors and Lists". At the bottom right is a slide number: "17".

Next, we will see how to concatenate the list. Two lists can be concatenated using the concatenation function. The syntax for that is concatenation of list 1 and list 2. We have a list which already contains three attributes, you want add another attribute which is employee dot ages; for that I am creating a new list which contains the ages of the employees five employees.

Now, I want to concatenate this new list that is m dot ages with the original list which is emp dot list. So, when you want to concatenate these two lists you have to use this concatenation operator, the original list and then the new list. So, this command concatenates these two lists, you are now assigning it to the employee dot list. When you print this new employee dot list you will see that you have added another attribute ages to the original list.

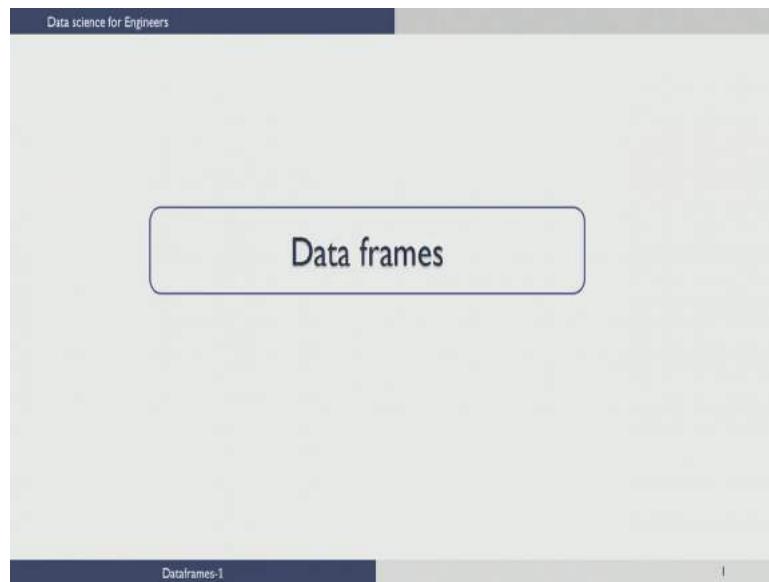
To summarize, we have seen the data types that are supported an R and two objects of R vectors and list in detail. In the next lecture, we are going to look at the important data object of R which is data frames.

Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

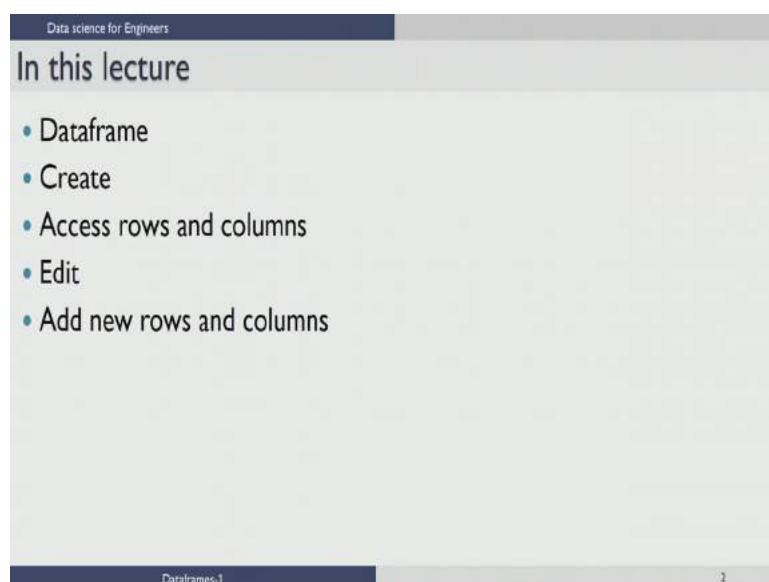
Lecture – 05
Data frames

(Refer Slide Time: 00:12)



Welcome to the lecture 4 in the module r of the course data science for engineers. In this lecture we are going to introduce you to the data frame objects of R.

(Refer Slide Time: 00:23)



How to create data frames, how to access rows and columns of data frame, how to edit data frames and how to add new rows and columns of the data frame. Let us first look at what data frame is?

(Refer Slide Time: 00:40)

The screenshot shows the RStudio interface. The title bar says "Data science for Engineers" and the main title is "Dataframes: Create dataframe". A callout box highlights the text "Data frames are generic data objects of R, used to store tabular data".

Code:

```
# Introduction to data frames
vec1 = c(1,2,3)
vec2 = c("R","Scilab","Java")
vec3 = c("For prototyping",
        "For prototyping","For Scaleup")
df = data.frame(vec1,vec2,vec3)
print(df)
```

Console Output:

```
> # Introduction to dataframes
> vec1 = c(1,2,3)
> vec2 = c("R","Scilab","Java")
> vec3 = c("For prototyping",
+         "For prototyping","For Sc
aleup")
> df = data.frame(vec1,vec2,vec3)
> print(df)
   vec1    vec2    vec3
1     1      R For prototyping
2     2  Scilab For prototyping
3     3    Java   For Scaleup
>
```

Data frame are generic data objects of R which you are used to store the tabular data. Data frames are the most popular data objects in R programming because we are comfortable in seeing the data in the tabular form. Data frames can also be taught as mattresses where each column of a matrix can be of different data type. Let us see how to create a data frame in R.

The code here shows how to create a data frame; the command here where the mouse is pointed creates a vector which is a numeric vector, which is containing 1, 2 and 3. The second command creates a character vector which contains 3 strings are Scilab and java; and the third command here is creating another character vector which is having entries for prototyping and first scale up.

The way you create the data frame is use the data dot frame command and then pass each of the elements you have created as arguments to the function data dot frame. This command will create a data frame d f when you print the data frame df, this is how the output looks. So, we can see that the names of the variables you have created are taken as columns and the entries in the each column are of the same data type; this is the condition which you need to be satisfying while creating a data frame.

(Refer Slide Time: 02:27)

Data science for Engineers

Create a dataframe using data from a file

- A dataframe can also be created by reading data from a file using the following command
 - `> newDF = read.table(path="Path of the file")`
- In the path, please use `\` instead of `\`
 - `> Example: "C:/Users/hii/Documents/R/R-Workspace/"`
- A separator can also be used to distinguish between entries. Default separator is space, '`,`'
 - `> newDF = read.table(file="path of the file", sep)`

Dataframes-1

Data frames can also be created by importing the data from text file to the way you have to do it is you have to use the function called read dot table and the syntax for that is you assign the data which your reading to a new data frame you have name that data from which you want to create and then read dot table takes in the argument the path of the file.

Let say you have data in a sum text file where the data is separated by spaces, you have to use this command read dot table and path = path of the file which from which you want import the data. This path specification is the os dependent you have to take care of whether you need to use backslash are slash operator depending upon your OS. A separator can also be specified to distinguish between entries the default separated between the entries of data is space, when you want to see the syntax for importing the data and creating a data frame this is how it looks; new data frame is read dot table you have to specify the path of the file and then you can also specify the separator that is being used to separate the entries of data.

The separator can also be a comma or a tab etcetera. So, what we have seen here is you can either create data frames on the go or you can use the data that is already existing in some format and use that to create data frames. Now that we have created a data frame,

(Refer Slide Time: 04:13)

The screenshot shows a RStudio interface. The top navigation bar says "Data science for Engineers". The main title is "Accessing rows and columns". Below the title is a list of bullet points:

- df[val1,val2] refers to row "val1", column "val2". Can be number or string
- "val1" or "val2" can also be array of values like "1:2" or "c(1,3)"
- df[val2] (no commas) - just refers to column "val2" only

Below this is a "Code" pane and a "Console Output" pane.

Code

```
# accessing first & second row:  
print(df[1:2,])  
# accessing first & second column:  
print(df[,1:2])  
# accessing 1st & 2nd column -  
# alternate:  
print(df[1:2])
```

Console Output

```
> print(df[1:2,])  
vec1 vec2      vec3  
1    1      R   For prototyping  
2    2 Scilab For prototyping  
> # accessing first & second column:  
> print(df[,1:2])  
vec1 vec2  
1    1      R  
2    2 Scilab  
3    3 Java  
> # accessing 1st & 2nd column - alternate:  
> print(df[1:2])  
vec1 vec2  
1    1      R  
2    2 Scilab  
3    3 Java
```

At the bottom left is a tab labeled "Dataframes-1". At the bottom right is a small number "5".

We need to see how we can access rows and columns of a data frame; the syntax for that is the data frame and 2 arguments has to be passed, the first argument val1 refers to the rows of a data frame and the second argument val2 refers to the columns of a data frame. So, this val1 and val2 can be array of values such as one to 2 or c of one comma, etcetera.

If you specify only val2 which is the syntax given here df of a 1 2 this refers to the set of columns, that you need to access from the data frame. In this code we can see that if you want to access first and second row of the data frame that is created. You can do so by accessing the rows we have to put 1 to 2 comma, nothing in the column specifies all the columns has to be accessed.

You can see in the result from the data frame what you have created in the previous slide you are able to access the first 2 rows. If you want to access the first 2 columns; instead of rows what you need to do is you need to leave a space first and then comma, and you need to specify the list of columns you need to access that is one to 2 that is shown here in this command and we can see the result on the console output, you are able to access the first 2 columns of the data from which you have created. If you want to access the first and second columns using just the column names you can do.

So, by just specifying d f of 1 to 2, this is another way of accessing the columns of a data frame.

(Refer Slide Time: 05:58)

The screenshot shows the RStudio interface with the following details:

Code:

```
# Data frame example 2
pd=data.frame("Name"=c("Senthil","Senthil","Sam","Sam"),
"Month"=c("Jan","Feb","Jan","Feb"),
"BS" = c(141.2,139.3,135.2,160.1),
"BP" = c(90,78,80,81))
pd2 = subset(pd,Name=="Senthil" | BS > 150 )
print("new subset pd2")
print(pd2)
```

Console Output:

```
> # Data frame example 2
> pd=data.frame("Name"=c("Senthil","Senthil","Sam","Sam"),
+ "Month"=c("Jan","Feb","Jan","Feb"),
+ "BS" = c(141.2,139.3,135.2,160.1),
+ "BP" = c(90,78,80,81))
> pd2 = subset(pd,Name=="Senthil" | BS > 150 )
> print("new subset pd2")
[1] "new subset pd2"
> print(pd2)
   Name Month  BS BP
1 Senthil Jan 141.2 90
2 Senthil Feb 139.3 78
4   Sam   Feb 160.1 81
```

Sometimes you will be interested in selecting the subset of the data frame; based on certain conditions. So, the way you do is you should have the conditions based on which you have to select the data frame and you should also have a data frame, once you have you can use the command subset to get the subset of data frame.

Let us illustrate by an example. Now we are going to create a data frame by name pd; using the first line which has name month blood sugar and blood pressure as the columns in the name we have Senthil and Sam in the month we have Jan and February, in the blood sugar we have a vector of blood sugar values and in the blood pressure you have a vector of blood pressure values you can print the data frame and see how this it looks.

But in that data frame what I want to extract is a subset where the name has to be Senthil or the blood sugar value has to be greater than 150. Now I can print the new data frame with this pd2 the result is as shown in the console output here. The original data frame contains all the entries, but the new data from pd2 selects these entries because the first entry is selected because the name is Senthil, the second entry is selected because the name is Senthil the third entry is selected because the blood sugar value is greater than 150.

(Refer Slide Time: 07:30)

The screenshot shows the RStudio interface. The title bar says "Data science for Engineers". The main area has a header "Editing dataframes" and a sub-header "Dataframes can be edited by direct assignment". Below this is a "Code" tab. The code pane contains the following R code:

```
# Introduction to dataframes
vec1 = c(1,2,3)
vec2 = c("R","Scilab","Java")
vec3 = c("For prototyping", "For prototyping","For Scaleup")
df = data.frame(vec1,vec2,vec3)
print(df)
df[2][2] = "R"
```

The console pane shows the execution of this code. It starts with the code itself, followed by the output of the print command, which is a data frame:

vec1	vec2	vec3
1	R	For prototyping
2	Scilab	For prototyping
3	Java	For Scaleup

Then, the assignment df[2][2] = "R" is shown, followed by the updated data frame where the entry "Scilab" has been replaced by "R".

Now we will see how to edit data frames; much like list you can edit the data frames by direct assignment. We have seen this data frame earlier we have vector 1, vector 2, vector 3 containing the elements in them we have created a data frame using this command. We can print that data frame also; now if I want to change the second entry in vector 2 as an R instead of Scilab I can achieve that by using this command I am accessing and I want to replace the element in the second row second column with the string R, when you execute this command d f of 2 comma = r what it does is it replaces the entry Scilab with R as shown in the results.

You can see that the Scilab has been replaced with the R, this is how you can edit the data frame by direct assignment.

(Refer Slide Time: 08:38)

The screenshot shows a slide titled "Editing dataframes". The slide content includes:

- A bulleted list:
 - A data frame can also be edited using the `edit()` command
 - Create an instance of data frame and use `edit` command to open a table editor, changes can be manually made
- A code block labeled "Code":

```
# Editing a data frame
myTable = data.frame()
myTable = edit(myTable)

English Maths Science
1     85   99    88
2     80   100   81
3     92   98    92
4     67   90    78
5     76   85    87
6     87   100   92
7     77   78    95
```
- An annotation with arrows pointing from the code to a data editor window. The text "Enter these values in the table" is above the arrow, and "And close the editor" is below it.
- A screenshot of the "Data Editor" window showing a table with columns English, Maths, and Science, containing the same data as the code.

Next, we see anything a data frame using edit command. So, what you need to do for this is you have to create an instance of data frame for example, you can see that I am creating an instance of data frame and naming it as my table by using the command data dot frame, this creates an empty data frame and I can use this edit command to edit the entries in my data frame. To do that what I have done is I am assigning whatever that is being edited into create a frame my table. So, when I execute this command it will pop up a window, where I can fill in the details what I want to fill in and then when I close this will save the data as a data frame by name my table.

Next, we will see how to add extra rows and columns to the data frame. We will continue the same example which we have used and now let us say we want to add another row to the data frame which you have created earlier.

(Refer Slide Time: 09:45)

The screenshot shows a RStudio interface. The title bar says "Data science for Engineers". The main area has a header "Adding extra rows and columns" with a note "Extra row can be added with "rbind" function and extra column with "cbind"". Below this, there are two panes: "Code" and "Console Output".

Code:

```
# continuing from previous example
# adding extra row and column:
df = rbind(df,data.frame(vec1=4,
vec2="C", vec3="For Scaleup"))
print("adding extra row")
print(df)
df = cbind(df,vec4=c(10,20,30,40))
print("adding extra col")
print(df)
```

Console Output:

```
> # continuing from previous example
> # adding extra row and column:
> df = rbind(df,data.frame(vec1=4,
+ vec2="C",
+ vec3="For Scaleup"))
> print("adding extra row")
[1] "adding extra row"
> print(df)
   vec1 vec2      vec3
1    1   R For prototyping
2    2   Scilab For prototyping
3    3   Java   For Scaleup
4    4   C    For Scaleup
> df = cbind(df,vec4=c(10,20,30,40))
> print("adding extra col")
[1] "adding extra col"
> print(df)
   vec1 vec2      vec3 vec4
1    1   R For prototyping 10
2    2   Scilab For prototyping 20
3    3   Java   For Scaleup 30
4    4   C    For Scaleup 40
```

At the bottom left is a navigation bar with icons for back, forward, and search. At the bottom center is the text "Dataframes-1". At the bottom right is the number "9".

So, we can add extra row using the command r bind and to add an extra column we use the command c bind. Let us see how we can add an extra row using r bind command the syntax for that is r bind, the data frame to which you want add and the entries for the new row you have to add you have to be careful when using r bind because the data types in each column entry should be = the data types that are already existing rows..

So, we are creating another row entry in which we have in the column 1 that is vec1 we have a numeric data type 4, in the column 2 we have a character variable c, in the column 3 we have a character variable for scale up. So, this command adds the row to the data frame and when you print the data frame you can see that row has been added to the original data frame.

Now, let us see how to add a column; adding a column is simple this can be done using a c bind command the syntax for that is c bind the original data frame and the entries for the new column. Now I am going to add a new column call vec4 which contains the entries 10 20 30 40 and when you add a new column to this and print the new data frame, you can see that this vec4 is added to the existing data frame.

(Refer Slide Time: 11:18)

The screenshot shows a R session titled "Data science for Engineers" with a slide overlay titled "Deleting rows and columns". The slide states: "There are several ways to delete a row/column, some cases are shown below". The code block contains the following R commands:

```
# continuing from previous example
# Deleting rows and columns:
df2 = df[-3,1]
print(df2)
# conditional deletion:
df3 = df[!names(df) %in% c("vec3")]
print(df3)
df4 = df[df$vec1==3,]
print(df4)
```

Annotations highlight specific parts of the code:

- A yellow arrow points to the line `df2 = df[-3,1]` with the text "A '-' sign before value and before ',' for rows & after ',' for columns".
- A yellow arrow points to the line `df3 = df[!names(df) %in% c("vec3")]` with the text "!' means no to those rows /columns which satisfy the condition".

The output of the R session shows the results of each command:

```
> print(df2)
  vec2      vec3 vec4
1  R For prototyping 10
2  Scilab For prototyping 20
4  C For scaleup 40
> # conditional deletion:
> df3 = df[!names(df) %in% c("vec3")]
> print(df3)
  vec1 vec2 vec4
1  R   10
2  Scilab 20
3  Java 30
4  C   40
> df4 = df[df$vec1==3,]
> print(df4)
  vec1 vec2      vec3 vec4
1  R   10
2  Scilab For prototyping 20
4  C   For scaleup 40
```

Now we will see how to delete rows and columns in a data frame there are several ways to delete rows and columns; we will see some of them to delete a row or a column, you need to access that row first and then insert a negative sign before that close, it indicates that you had to delete that rows. So, let us see the example here now from the data frame we have if you want to delete the third row and the first column that can be done using this command. So, I want to delete third row so I chose the third row and insert a negative symbol before it.

Similarly, I want to delete a column one I chose that column and then insert a negative symbol before that column. And I am assigning that to new data frame df2 now when I print the df2. You can see in the results we do not have the column vector one and we do not have vector 3 which is what we expected to happen. We can also do conditional deletion of rows and columns as we have seen this command will delete column 3 from the data frame what we have created.

So, the explanation goes as follows; we have a data frame we want to access all the rows in the columns we want to access those columns where there is no vector 3; that means, we want to access vector 2 vector 4 and vector one. So, this exclamatory symbol says no to the columns that are having column name vector 3 and I am assigning that to data from 3 when I print data from 3 you can see that there is no column vec3 in the data

frame which we are looking for, you can also delete the rows where we have an entry 3 by using this command.

So, what you are saying here is access those rows where the element in the vector 1 is not = 3 and we need to access all the columns. So, and we are assigning that 2 data from df df4 and when you print the df4 we can see that the row which is having the entry 3 is deleted from the data frame.

(Refer Slide Time: 13:35)

Data science for Engineers

Manipulating rows – the factor issue

- When character columns are created in a data.frame, they become factors
- Factor variables are those where the character column is split into categories or factor levels

Code	Console Output
# Manipulating rows in data frame # continued from previous page df[3,1]=3.1 df[3,3]="Others" print(df)	> # Manipulating rows in data frame > # continued from previous page > df[3,1]=3.1 > df[3,3]="Others" warning message: In [-> factor ("tmp", iseq, value = "Others") : invalid factor level, NA generated > print(df) vec1 vec2 vec3 1 1.0 R For prototyping 2 2.0 Scilab For prototyping 3 3.1 Java NA

Notice the NA values displayed instead of the string "Others".
Also see the use of the word "factor" in the warning above

Dataframes-1

now we will see how to manipulate the rows in the data frame and what is called as a factory issue. R has inbuilt characteristic to assign the data types to the data you enter. When you enter numeric variables, it knows all the numeric variables that are available when you enter character variables it takes whatever the character variables you are giving as categories or factors levels.

And it assumes that these are the only factors that are available for now; when you want to change the element in the third row third column to others; what happens is it will display warning message saying that, this others categorical variable is not available and it replaces that with the NA you can notice that the place where we want others to be there we are having a NA and we can also see the use of word factor in the warning message, how to get rid of the factor issue is the question now.

(Refer Slide Time: 14:43)

The screenshot shows a RStudio interface. The title bar says "Data science for Engineers". The main area has a header "Resolving factor issue" with a sub-instruction: "New entries need to be consistent with factor levels which are fixed when the dataframe is first created". Below this is a "Code" section containing R code, and a "Console Output" section showing the results of running that code.

Code

```
vec1 = c(1,2,3)
vec2 = c("R","Scilab","Java")
vec3 = c("For prototyping",
        "For prototyping","For Scaleup")
df = data.frame(vec1,vec2,vec3,
                stringsAsFactors = F)
# Now trying the same manipulation
df[3,3] = "Others"
print(df)
```

Console Output

```
> vec1 = c(1,2,3)
> vec2 = c("R","Scilab","Java")
> vec3 = c("For prototyping",
+         "For prototyping","For Scaleup")
> df = data.frame(vec1,vec2,vec3,
+                  stringsAsFactors = F)
> # Now trying the same manipulation
> df[3,3] = "Others"
> print(df)
   vec1    vec2      vec3
1     1      R For prototyping
2     2  Scilab For prototyping
3     3      Java       Others
>
> |
```

New entries in R when you are entering should be consistent with the factor levels that are already defined if not those error message will be printed out. If you do not want this issue to happen what you have to do is while defining the data from itself you need to pass another argument, which says strings as factors is false by default this argument is true that is the reason why you get this warning message when you want to change the string characters into new string characters as an element..

Now try doing the same manipulation you want to change the third row third element to others and print the data frame you can see that there is no NA anymore and we achieved what we want. In this lecture we have seen how to create data, frames how to access rows and columns of data frame and how to delete rows and columns of a data frame and so on.

In the next lecture we are going to see some other operations that can be done on data frames.

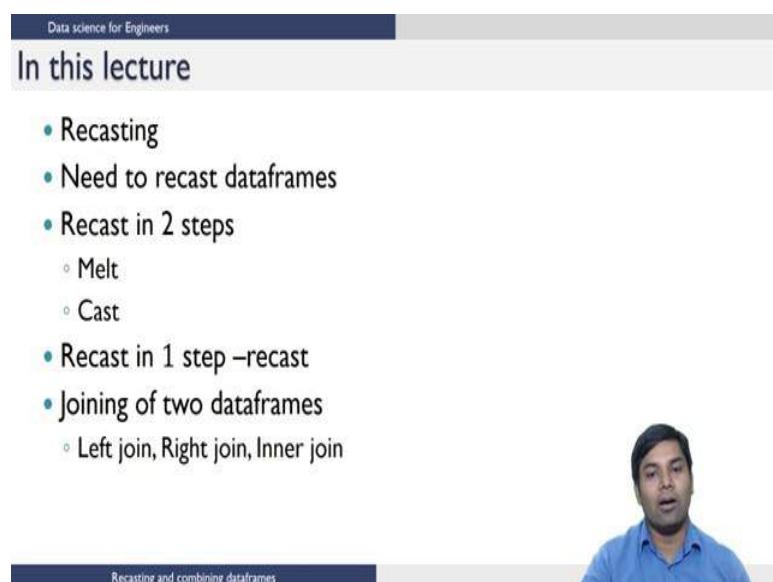
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 06
Recasting and joining of dataframes

Welcome to the lecture 5, in the r module of the course data science engineers. In the previous lectures, we have seen how to create data frames, How to access rows and columns of data frames, How to add rows and columns to existing data frame. And so on. Here we will look at more sophisticated operations on data frames, such as recasting and joining of data frames.

(Refer Slide Time: 00:42)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header is a light gray section with the title 'In this lecture' in bold black font. The main content is a bulleted list of topics:

- Recasting
- Need to recast dataframes
- Recast in 2 steps
 - Melt
 - Cast
- Recast in 1 step –recast
- Joining of two dataframes
 - Left join, Right join, Inner join



In this lecture we are going to first define, what is recasting of a data frame mean? Why do one need to recast the data frames, How the recasting can be done in 2 steps using melt and cast command, How the recasting can be done in a single step using recast command and how to join 2 data frames using left join right join and inner join functions of d player package in r.

(Refer Slide Time: 01:10)

Data science for Engineers

Recasting dataframes

• Recasting is the process of manipulating a data frame in terms of its variables

• Reshaping the data

• insights

Dataframe – “pd”

	Name	Month	BS	BP
1	Senthil	Jan	141.2	90
2	Senthil	Feb	139.3	78
3	Sam	Jan	135.2	80
4	Sam	Feb	160.1	81

variable Month Sam Senthil

	variable	Month	Sam	Senthil
1	BS	Feb	160.1	139.3
2	BS	Jan	135.2	141.2
3	BP	Feb	81.0	78.0
4	BP	Jan	80.0	90.0

Recasting and combining dataframes

Let us first see what is recasting of data frames means, requesting as a process of manipulating data free in terms of it is variables. Why do want wants to recast the data frames? The answer is recasting helps in reshaping the data which could bring more insights on the data, when it is seen from the different perspective. Let us take a data frame which is created in the last lecture, we have the data frame name pd which has column name month, blood sugar and blood pressure. So now, you want to convert this data frame into the other form which is shown below, where you have blood sugar and blood pressure as the variables of importance to you and this involves an operation which is called recasting, this recasting is demonstrated using an example here.

(Refer Slide Time: 02:01)

```
Data science for Engineers
Recast in two steps: Example
• Create the following example : data frame 'pd'


|   | Name    | Month | BS    | BP |
|---|---------|-------|-------|----|
| 1 | Senthil | Jan   | 141.2 | 90 |
| 2 | Senthil | Feb   | 139.3 | 78 |
| 3 | Sam     | Jan   | 135.2 | 80 |
| 4 | Sam     | Feb   | 160.1 | 81 |


Code
# Data frame example 2
pd=data.frame("Name"=c("Senthil",
"Senthil","Sam","Sam"),
"Month"=c("Jan","Feb","Jan","Feb"),
"BS" = c(141.2,139.3,135.2,160.1),
"BP" = c(90,78,80,81))
print(pd)
Console Output
> pd=data.frame("Name"=c("Senthil",
"1","Senthil","Sam","Sam"),
+ "Month"=c("Jan",
"Feb","Jan","Feb"),
+ "BS" = c(141.2,1
39.3,135.2,160.1),
+ "BP" = c(90,78,8
0,81))
> print(pd)
  Name Month BS BP
1 Senthil Jan 141.2 90
2 Senthil Feb 139.3 78
3 Sam Jan 135.2 80
4 Sam Feb 160.1 81
>

```

Recasting and combining dataframes



So, in order to do recasting we have to have a data frame, which is the following which is shown in screen. To create this data frame, you can use the code that is displayed in screen this one and when you use this code and execute this, you will see the data frame which is shown here. Since we have the data frame now, we can see how to recast the existing data frame into the form which we want.

(Refer Slide Time: 02:36)

```
Data science for Engineers
Recast in two steps: Example
• Two steps
  • Melt
  • Cast
• Identifier (Discrete type variables)
• Measurements (numeric variables)
• Categorical and Date variables can not be measurements
```

Diagram illustrating the structure of a data frame:

	Name	Month	BS	BP
1	Senthil	Jan	141.2	90
2	Senthil	Feb	139.3	78
3	Sam	Jan	135.2	80
4	Sam	Feb	160.1	81

Identifier variables → Name, Month
measurement variables → BS, BP

Recasting and combining dataframes



Let us see an example to demonstrate this, how to recast the data frame into another form, using 2 steps first one is melt and the second one is cast. This is the data from you

have when you want to use melt and cast command to recast, the data frame you need to identify what are called identifier variables and measurement variables of your data frame. The rules for identifying this identifying variables are, most of the discrete type variables can be identifier variables, measure the numeric variables can be measurement variables and there are certain rules for the measurement variables such as, categorical and date variables cannot be measurement variables. So, the key idea is from the data frame, you have to identify what are called identifier variables? And what are called measurement variables? Once you have identify this identifier variables and measurement variables, you are ready to do the melt operation which you are going to see now.

(Refer Slide Time: 03:28)

The screenshot shows a slide titled "Step 1: Melt" under the heading "Data science for Engineers". The slide content includes a code block and a "Console Output" window.

Code:

```
# Data frame example 3
# melt operation sample code
library(reshape2)
Df = melt(pd, id.vars = c("Name","Month"),
          measure.vars = c("BS", "BP"))
print(Df)
```

Console Output:

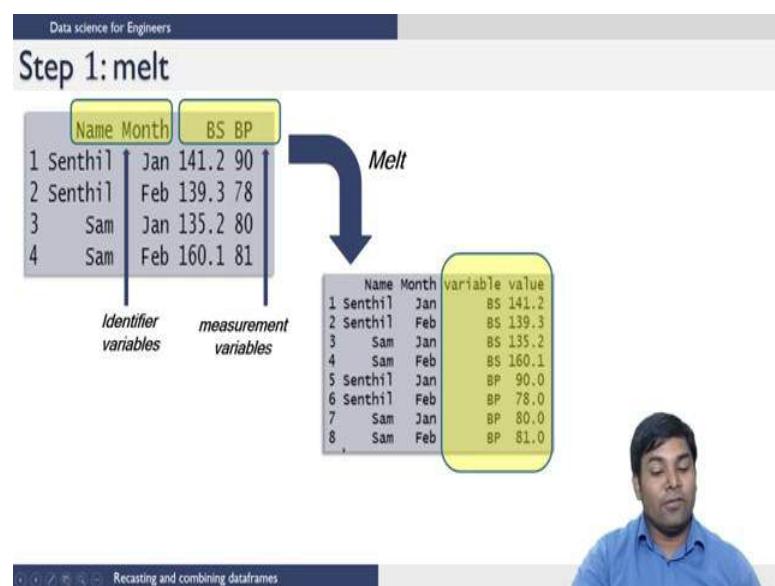
```
> # Data Frame example 3
> # melt operation sample code
> library(reshape2)
> Df = melt(pd,
+ id.vars = c("Name", "Month"),
+ measure.vars = c("BS", "BP"))
> print(Df)
   Name Month variable value
1 Senthil Jan       BS 141.2
2 Senthil Feb       BS 139.3
3 Sam     Jan       BS 135.2
4 Sam     Feb       BS 160.1
5 Senthil Jan       BP  90.0
6 Senthil Feb       BP  78.0
7 Sam     Jan       BP  80.0
8 Sam     Feb       BP  81.0
```

This melt command is available in the reshape2 library, here is the first time we are loading another library to perform some operations. In the pre-course material we have given you how to install packages and this library command helps you do load the packages, that are already installed and this is the syntax of the melt comment. For the melt command you have to give the data frame as first argument and you have to specify what are the identifier variables in your data frame , and you have to also specify what are the measurement variables in your data frame and these are the default variable and value arguments that, are generated when the melt command is executed. To do this melting operation, we can use this code you first have to load this library reshape 2 which contains this functions melt and cast.

So, the syntax we have seen, melt you need to pass the data frame which you want to melt and you have to also specify, what are the identifier variables in the data frame you pass, what are the measurement variables that you are passing? Once you do this initial data frame will become like this, what it has done is, since this name and month or given as Id variables, there as it is and measurement variables BS and BP are now start under a column by name variable, as you can see here and the values of them are stored in another column, which is named as value.

So, when the melt command is executed, it will take this Id variables and keep them intact and then convert the measure variables into, one single column which is given by variable and stores the values of those variables, in another column by name value, that is what when you say in this syntax variable dot name is variable and value dot name is value means.

(Refer Slide Time: 05:34)



So, the columns which carries this measurement variables is named as variable and which wholes the values of the measurement variable is named as value. So, this is the first step identifier variables and measurement variables of the data frame and uses the melt command to melt the data frame to get to this structure.

(Refer Slide Time: 05:44)

The slide is titled "Step 2: cast". It contains a list of bullet points:

- Applying the dcast() function
- dcast (data, formula, value.var = col. with values)

The "Code" section shows the following R code:

```
# cast operation sample code
# continued from previous code
# we use dcast as we are working on
# a data frame
Df2 = dcast(Df,
             variable+month ~ Name,
             value.var="value",
             print(Df2))
```

The "Console Output" section shows the resulting data frame:

	variable	Month	Sam	Senthil
1	BS	Feb	160.1	139.3
2	BS	Jan	135.2	141.2
3	BP	Feb	81.0	78.0
4	BP	Jan	80.0	90.0

Annotations explain the code:

- A red box highlights the formula part: `variable+month ~ Name`. A yellow arrow points to it from the text "Column of Df from which the values are to be taken from".
- A red box highlights the `value.var="value"` part. A yellow arrow points to it from the text "Columns "variable" & "month" to remain as is."
- A red box highlights the `print(Df2)` part. A yellow arrow points to it from the text "Categories in column "Name" become new variables."

A small video thumbnail of a man speaking is visible on the right.

Next step is the cast, since we are using data frame here, we use the function d cast this d cast function is also available in reshape 2 library the syntax for d cast is as follows, the d cast command takes in the data frame, which you want to d cast and the formula which will explain for this case what it is? And value dot var. So, you have to specify the columns from which the values to be taken from when you are d casting.

Let us see the example our case, here you have a data frame Df which you already melted. Now, you are creating another data frame Df2 by using d cast command, this is the data frame which you are passing that is Df and this is the formula. What does it say? I want to have this variable and month as constant, because you want blood sugar and blood pressure to be your variables of importance and then, you have to convert the name variable into 2 columns are, how many of a columns depending upon the number of categories in the name.

That is what this formula explains, columns variable and month remain as it is and the categories in the name becomes new variable. We have 2 categories in this example, which are Sam and Senthil and they become the new columns, that are new variables and the values for those variables has to be picked from the column value, that is what this value dot variable suggests. Once this operation is done, if you print the data frame this is how you will get the data frame in your required format.

(Refer Slide Time: 07:15)

Data science for Engineers

Step 2: cast

```
Df2 = dcast(Df, variable+month ~ Name, value.var="value")
```

	variable	Month	Sam Senthil
1	BS	Feb	160.1 139.3
2	BS	Jan	135.2 141.2
3	BP	Feb	81.0 78.0
4	BP	Jan	80.0 90.0

	Name	Month	variable	value
1	Senthil	Jan	BS	141.2
2	Senthil	Feb	BS	139.3
3	Sam	Jan	BS	135.2
4	Sam	Feb	BS	160.1
5	Senthil	Jan	BP	90.0
6	Senthil	Feb	BP	78.0
7	Sam	Jan	BP	80.0
8	Sam	Feb	BP	81.0

Recasting and combining dataframes

So, you have this melted data frame from this, when you apply dcast function you pass in this data frame and you say variable and month are the ones, which you want to have it as constant, that are the left side of the formula and in the to the right of the formula you have name. So, in this name column we have 2 categories Sam and Senthil and those will be created as 2 new columns and the values for those columns, have to be taken from the value column of the melted data frame, that is how the cast command works.

(Refer Slide Time: 07:52)

Data science for Engineers

Recasting in single step

- Applying the recast() function performs melt and cast in one command
- recast(data, formula, ..., id.var, measure.var)

Command & console Output

Parameter refers to the "cast" section of the command Parameter refers to the "melt" section of the command

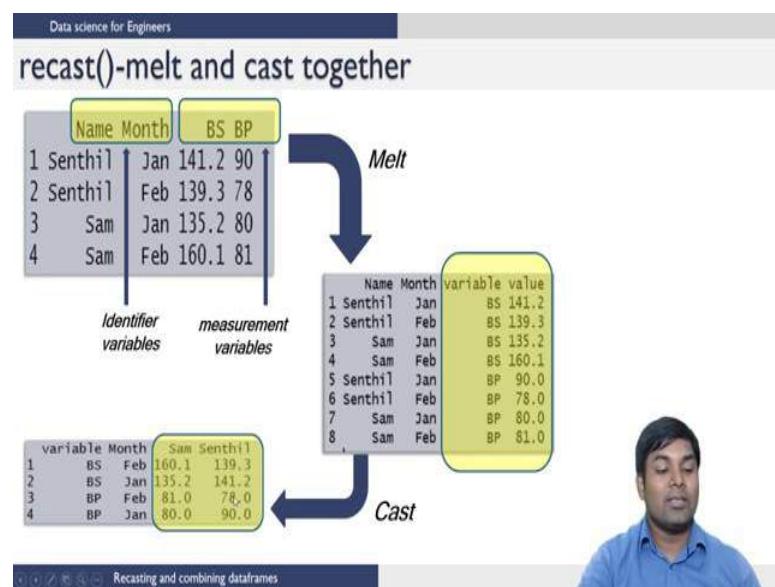
```
Console -> 
> recast(pd, variable+Month~Name, d, var=c("Name", "Month"))
variable Month Name   value
1      BS    Feb  Sam  139.3
2      BS    Jan  Senthil 141.2
3      BP    Feb  Sam   78.0
4      BP    Jan  Senthil  90.0
>
```

Recasting and combining dataframes

Now, let us see how to do this recasting in a single step. So, recasting can perform in a single step, using recast function the syntax for this is as follows, recast you have to give the data and the formula and we have to also give id variables and measurement variables. So, if you can see these input arguments, it takes the input arguments of both melt and cast as you can see in this command here.

So, recast command takes the data frame and it also takes the formula, this is the parameter that refers to the cast section of the command and this is the parameter, that refers to the melt section of the command. What we have seen in the melt, we have to specify what are the Id variables and the measurement variables. So, when you specify only Id variables, the rest of the variables are defaultly taken as the measurement variables. So, that is why we did not specify measurement variables here, you can also specify the measurement variables, as we can see from the syntax. Now when you execute this command, it will melt and it also cast and it will print the casted data frame as shown in this screen below. Next with this, we can see that melt and cast operations can be done together using the recast command.

(Refer Slide Time: 09:05)



(Refer Slide Time: 09:16)

Data science for Engineers

Add new variable to dataframe based on existing ones

- Call the library 'dplyr' command using the library() command
- mutate() command will add extra variable columns based on existing ones.

Code	Console Output																														
<pre># Adding new variables #Continue from #example on slide 3 library(dplyr) pd2 <- mutate(pd, log_BP = log(BP)) print(pd2)</pre>	<pre>> # Adding new variables > #Continue from > #example on slide 3 > library(dplyr) > pd2 <- mutate(pd, log_BP = log(BP)) > print(pd2)</pre> <table border="1"><thead><tr><th></th><th>Name</th><th>Month</th><th>BS</th><th>BP</th><th>log_BP</th></tr></thead><tbody><tr><td>1</td><td>Senthil</td><td>Jan</td><td>141.2</td><td>90</td><td>4.499810</td></tr><tr><td>2</td><td>Senthil</td><td>Feb</td><td>139.3</td><td>78</td><td>4.356709</td></tr><tr><td>3</td><td>Sam</td><td>Jan</td><td>135.2</td><td>80</td><td>4.382027</td></tr><tr><td>4</td><td>Sam</td><td>Feb</td><td>160.1</td><td>81</td><td>4.394449</td></tr></tbody></table>		Name	Month	BS	BP	log_BP	1	Senthil	Jan	141.2	90	4.499810	2	Senthil	Feb	139.3	78	4.356709	3	Sam	Jan	135.2	80	4.382027	4	Sam	Feb	160.1	81	4.394449
	Name	Month	BS	BP	log_BP																										
1	Senthil	Jan	141.2	90	4.499810																										
2	Senthil	Feb	139.3	78	4.356709																										
3	Sam	Jan	135.2	80	4.382027																										
4	Sam	Feb	160.1	81	4.394449																										

- original data frame 'pd' is the first argument
- multiple variables can be created as transformation of old variable
- here, new variable column is "log_BP" which is log of variable column "BP"

Recasting and combining dataframes

Next, we see how to create a new variable, that is a function of already existing variable, using the mutate command. Sometimes it is essential to have a translated or the function are variable, which is created from the existing variables. In this case, let us assume logarithm of BP value is something which is giving us more insight about the data. How do you create a new variable which carries the logarithm value of the blood pressure from the existing blood pressure value-is the question.

So now, how to do that is you have to load the library dplyr, you can use mutate command and you need to pass the data frame and you have to say, you have to create new column which is carrying the values of logarithm the existing column BP. Now, if you print this pd2 you can see that, there is another variable that is logarithm of BP that is created and you have the corresponding values of it. Now, let us look how to join 2 data frames it is very important in terms of data analysis, because you will get part of the data from one source and the part of the data from other source, when we want to match these 2 data, which are having some common IDs, how do you do this-is the question.

(Refer Slide Time: 10:33)

Data science for Engineers

Combining two dataframes – dplyr package

The common syntax for "dplyr" functions used to combine dataframes:
`"function(dataframe1, dataframe2, by = id.variable)"`

- The "id.variable" is common to both dataframes
- This variable provides the identifiers for combining the 2 dataframes
- The nature of combination depends on the function to be used
- Illustration Example : A possible combination

The diagram shows two dataframes being combined. On the left, a datafarme with columns ID, Name, and Age has its ID column circled. An arrow points from this circled column to the ID column of another datafarme on the right, which also has columns ID, Name, and Age. The second datafarme's ID column is also circled. An arrow points from the plus sign between the two dataframes to the text 'id.variable "ID" is used to combine both dataframes column wise'. To the right of the combined datafarme, there is a small image of a man in a blue shirt.

id.variable "ID" is used to combine both dataframes column wise

Recasting and combining dataframes

So, this combining of data frames can be done using, dplyr package the general syntax of the dplyr is as follows, you need to have a function which could be either left join, right join, inner join and so on. And you need to pass the first data frame and the second data frame, because you want to do joining of this 2 data frames and you have to specify, by which I d variable you have to join this 2 data frames.

So, here the I d variable is common to both data frames; that means, you have to have that variable in both data frames, which you want to combine and this variable provides the identifiers for combining the 2 data frames and the nature of combination depends upon the function that is being used. We will see some examples and (Refer Slide Time: 11:23) example let us see this one, we have one data frame which carries I d name and age. We have one and 2 as I ds here, name as Jack and Jill whose ages are 10 and 12 at a suppose, we have another data frame which has his Ids in the reverse order, Id2 Id1 and gender is girl and boy and this is output you want to get, let us say you want to merge these 2 data frames using some function either left join or right join are something.

So, that you will get the data frame which contains information in both the individual data frames for example, you can see the I ds are the common variables or the identifiers variables that are common to both data frames and we are using this Id variable, to combine this 2 data frames 1 and 2. So, we have 1 Jack and for 1 we have boy and we have age of Jack as 10 and that is also been taken care, and you have Jill and the Id

variable of Jill is 2. So, we will have 2 Jill age and the gender this is one example, how the merging and combining the data frames happens? Now, let us look deep into the different functions, that available in the dplyr package to combine two data frames.

(Refer Slide Time: 12:39)

The slide has a dark blue header bar with the text 'Data science for Engineers'. The main title 'Combining two dataframes' is centered in a light gray header area. Below the title is a white content box with a thin black border. Inside the box, there is a bulleted list of instructions and a list of join functions. At the bottom right of the slide, there is a small video frame showing a man in a blue shirt speaking.

- Call the library 'dplyr' command using the library() command
- The following commands would be used to combine datasets:
 - ❖ left_join()
 - ❖ full_join()
 - ❖ right_join()
 - ❖ semi_join()
 - ❖ inner_join()
 - ❖ anti_join()

Recasting and combining dataframes

There are several functions that are available in the dplyr package to combine data frames, few of them are left join, right join and inner join and there are full join, semi join and anti-join. In this lecture, what we have going to see are the first 3 left join, right join and inner join. We will leave the audience as an exercise, to understand what full join, semi join and anti-join does in combining the data frames.

(Refer Slide Time: 13:08)

Data science for Engineers

Example: create first dataframe

Create the data frame 'pd'

	Name	Month	BS	BP
1	Senthil	Jan	141.2	90
2	Senthil	Feb	139.3	78
3	Sam	Jan	135.2	80
4	Sam	Feb	160.1	81

Code

```
# Data frame example 2
pd=data.frame("Name"=c("Senthil",
+Senthil","Sam","Sam"),
+ "Month"=c("Jan","Feb"),
+ "BS" = c(141.2,139.3,135.2,160.1),
+ "BP" = c(90,78,80,81))
print(pd)
```

Console Output

```
> pd=data.frame("Name"=c("Senthil",
+ "Senthil","Sam","Sam"),
+ "Month"=c("Jan",
+ "Feb","Jan","Feb"),
+ "BS" = c(141.2,139.3,135.2,160.1),
+ "BP" = c(90,78,80,81))
> print(pd)
#> #> #> #>
```

Recasting and combining dataframes

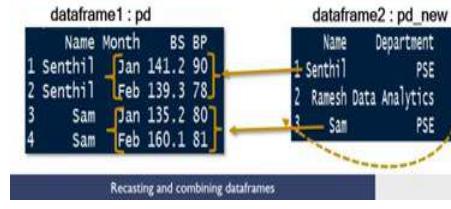


Let us illustrate joining of data frames by creating 2 data frames first, let us first create this data frame p d, this can be created using the code shown here and when you print that, you can see the output as share we have 2 names Senthil and Sam, we have 2 months Jan and February, we have blood sugar and blood pressure values of those variables. Now, we are taking another data frame which contains 3 names Senthil Ramesh and Sam and the other column carries the department, where they are working. So, Senthil and Sam is working in PSE and Ramesh is working in data analytics, to create this data frame you can use this code and when you print this data frame, you can see the result in the console has shown below. Now we have created 2 data frames pd and pd new.

(Refer Slide Time: 14:01)

left_join()

- joins matching rows of "dataframe2" to "dataframe1" based on the "id.variable"
- In the example, only "Sam" and "Senthil" from id.variable "Name" are present in "pd" which is dataframe1.
- Only these two IDs & corresponding values in "pd_new" will be merged with "pd"
- The variable "Department" from "pd_new" would be merged to pd



Let us look at how left join works. When you want to combine this 2 data frames pd and pd new a left join joins, matching rows of data frame 2 to the data from 1 based on the Id variables. From the syntax we can see that function data frame 1 and data frame 2 is the syntax we have and you have to specify the Id variable as the last argument. So now, if you want to left join data frame 1 which is pd and data frame 2 which is pd_new, what it takes as a reference is, the data frame 1 which is pd and now it matches the rows of the data frame 2, which is Senthil Ramesh and Sam and sees in the data frame 2, what is matching with the name variables in the data frame 1, essentially it will keep only Senthil and Sam and not keep Ramesh, because it will take to matching rows from the data frame 2 to the data frame 1.

So, well see that when you do the example, now here there are only 2 Ids corresponding to the values in pd_new and that will be merged with pd, the variable department will be added to the final data frame, only for Senthil and Sam.

(Refer Slide Time: 15:23)

The screenshot shows a slide titled "left_join()" from a course on Data science for Engineers. The slide includes the following elements:

- USE DATAFRAMES 'pd' and pd_new**: A callout box containing the command.
- Code**: A section containing the R code for performing a left join:

```
#using left_join()  
#to combine two dataframes  
#Continue from  
#example  
library(dplyr)  
pd_left_join1 <- left_join(pd, pd_new, by = "Name")  
print(pd_left_join1)
```

- dataframe1 : pd**: A table showing data for Senthil and Sam across January and February.
- dataframe2 : pd_new**: A table showing department information for Senthil, Ramesh, and Sam.
- pd_left_join1**: The resulting data frame after the join operation, combining the data from both frames.

Let us see in detail, you have 2 data frames you need to load the library dplyr and you are doing it, a left join and I am naming the new data frame, which is coming out with this left join operation as, pd underscore left underscore join 1. I have to use this command left join, this is the data frame 1 I am passing in and this is the data from 2 I am passing in and then I want to join this 2 data frames by the variable name .

Now, when you specify I want to join this 2 data frames by name, the left join it will take pd as a reference, look for the names that are common in both p d and p d new and then take the data from the pd_new, and merge it with the p d and then create another data frame, which is given by this name pd_left_join1. So, you have data frame 1 p d which contains, Senthil and Sam and it look for, Senthil and Sam in the data frame 2 and then merges the information, that is available extra for these names and add it to the existing data frame, with another column department and the department of Senthil is PSC, in the department of Sam is also PSC, it will rehold the pd and then add the corresponding piece of information, that is coming from the data frame 2.

(Refer Slide Time: 17:00)

```
Data science for Engineers  
right_join()  
Joins matching rows of "dataframe1" to "dataframe2" based on the "id.variable"  
  
Code  
#using right_join() #using  
right_join()  
#to combine two data frames  
#Continue from  
#example  
pd_right_join1 <- right_join  
(pd, pd_new, by = "Name")  
print(pd_right_join1)  
  
dataframe1 : pd  
  Name Month BS BP  
1 Senthil Jan 141.2 90  
2 Senthil Feb 139.3 78  
3 Sam Jan 135.2 80  
4 Sam Feb 160.1 81  
  
dataframe2 : pd_new  
  Name Department  
1 Senthil PSE  
2 Ramesh Data Analytics  
3 Sam PSE  
  
pd_right_join1  
  Name Month BS BP Department  
1 Senthil Jan 141.2 90 PSE  
2 Senthil Feb 139.3 78 PSE  
3 Ramesh <NA> NA NA Data Analytics  
4 Sam Jan 135.2 80 PSE  
5 Sam Feb 160.1 81 PSE
```

Recasting and combining dataframes

Now, let us look at the right join similarly. So, what right join does is, it joins matching rows of data frame 1 to the data frame 2 based on the Id variable. Let us say, you have this data from 1 which is pd and data frame 2 which is pd new and you can do the right join, by using right join command and you need to pass what is data frame 1 and what is data frame 2, here we have pd as a data frame 1 and pd new as a data frame 2. Now, what is it take is it will take the pd new as the reference data frame and try to match the rows, which are present in the pd new and look for a match in the pd. We have Senthil and Sam there are matching in the pd also and it will keep this Ramesh now, because the references is this data frame. So, you will have Senthil Ramesh and Sam, but for Ramesh you do not have month, blood sugar and blood pressure values, which are replace by n s when the matching operation is that.

(Refer Slide Time: 18:10)

The slide has a header 'Data science for Engineers' and a title 'right_join()'. A subtitle below the title reads 'Joins matching rows of "dataframe1" to "dataframe2" based on the "id.variable"'. The main content is divided into two sections: 'Code' and 'dataframe1 : pd_new'.

Code:

```
#using right.join() #using  
right_join()  
#to combine two data frames  
#Continue from  
#example  
pd_right_join2 <- right_join  
(pd_new, pd,  
by = "Name")  
print(pd_right_join2)
```

dataframe1 : pd_new

	Name	Month	BS	BP
1	Senthil	Jan	141.2	90
2	Senthil	Feb	139.3	78
3	Sam	Jan	135.2	80
4	Sam	Feb	160.1	81

dataframe2 : pd

	Name	Department
1	Senthil	PSE
2	Ramesh	Data Analytics
3	Sam	PSE

pd_right_join2

	Name	Department	Month	BS	BP
1	Senthil	PSE	Jan	141.2	90
2	Senthil	PSE	Feb	139.3	78
3	Sam	PSE	Jan	135.2	80
4	Sam	PSE	Feb	160.1	81

A yellow arrow points from the 'pd_right_join2' code line to the resulting table.

You can change the order, in which you pass the data frames and you can see that, if you change the order, you pass the data frame one has pd new and data frame 2 as pd. You can observe that output is similar to the left join, because now the reference variable here is pd, when you are using pd as a reference data frame even though you are doing this right join operation, the operation is similar to left join because your pd is the reference at the right join here.

So, to summarize left join and right join can be used vice versa, but depending upon the way you pass this data frames, the matching operations will either look similar or different. So, you have to be careful when you are passing the arguments, to this left and right join commands.

(Refer Slide Time: 19:09)

The slide title is "inner_join()". A subtitle says "Merges and retains those rows with IDs present in both dataframes".

Code:

```
#using inner_join()
#to combine two data frames
#Continue from
#example
library(dplyr)
pd_inner_join1 <- inner_join
(pd_new, pd, by = "Name")
print(pd_inner_join1)
```

dataframe1 : pd_new

Name	Department
1 Senthil	PSE
2 Ramesh	Data Analytics
3 Sam	PSE

dataframe2 : pd

Name	Month	BS	BP
1 Senthil	Jan	141.2	90
2 Senthil	Feb	139.3	78
3 Sam	Jan	135.2	80
4 Sam	Feb	160.1	81

pd_inner_join1

Name	Department	Month	BS	BP
1 Senthil	PSE	Jan	141.2	90
2 Senthil	PSE	Feb	139.3	78
3 Sam	PSE	Jan	135.2	80
4 Sam	PSE	Feb	160.1	81

A yellow arrow points from the highlighted line of code to the resulting data frame.

Now, let us see what inner join does, inner join merges and retains those rows in the ids present in the both data frames. Now you have data frame 1 which is pd new you have data frame 2 which is pd, now when I pass these 2 data frames as an argument to this inner join function and I want to match them, by name it will look for the rows with Ids present in the both data frames. In this 2 data frames we have Senthil and Sam present, it will print only the data that is corresponding to the Senthil and Sam, because Ramesh is not available in this data frame 2.

(Refer Slide Time: 19:54)

The slide title is "Combining two dataframes: summary".

left_join()	✓
right_join()	✓
inner_join()	✓
full_join()	✗
semi_join()	✗
anti_join()	✗

A yellow arrow points from the checkmarks in the first three rows to the checkmarks in the last three rows.

So, we have seen left join, right join and inner join. We left as an exercise for the viewer, to understand how full join semi join and anti-works. To summarize in this lecture, we have seen how to recast the data frames? And how to combine 2 data frames using the dplyr package? In the next lecture we are going to see how to do arithmetic logical and matrix operations in r.

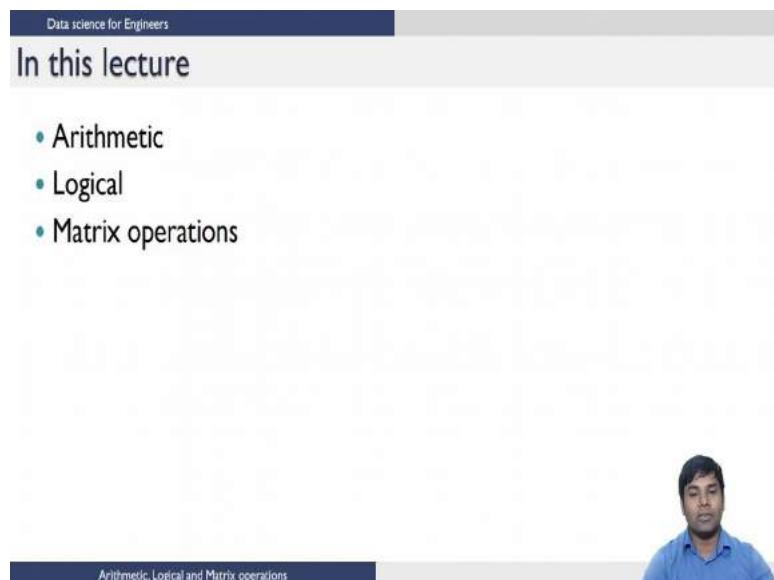
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 07
Arithmetic, Logical and Matrix operations in R

Welcome to lecture 6 of the R module in the course Data Science for Engineers. In the previous lectures we have seen various data types of R, how to access R delete the elements of the different data types and so on. Now, it is time to see how to perform arithmetic, logical and matrix operations in R.

(Refer Slide Time: 00:42)



The image shows a screenshot of a presentation slide. At the top, there is a dark blue header bar with the text "Data science for Engineers". Below this, a light gray bar contains the title "In this lecture". The main content area is white and lists three topics in a bulleted format:

- Arithmetic
- Logical
- Matrix operations

At the bottom of the slide, there is a dark blue footer bar with the text "Arithmetic, Logical and Matrix operations". To the right of this bar, there is a small video thumbnail showing a man in a blue shirt.

In this lecture we are going to see how to do arithmetic operations, logical operations and matrix operations in R.

(Refer Slide Time: 00:50)



So, let us first look at the arithmetic operations.

(Refer Slide Time: 00:54)



Symbols	Operation
=, <-	Assignment
+	Addition
-	Subtraction
*	Multiplication
/	Division
^, **	Exponent
%/%	Remainder
%/%	Integer division

* In R only <- is valid for assignment operation where as in R Studio both = and <- will work



R supports all the basic arithmetic operation, the first one is assignment operator. You can use either = or the back arrow <- to assign a value to be variable and standard addition, subtraction, multiplication, division, integer division and remainder operations are also available in R. In R back arrow <- is only the valid assignment operator whereas, as an R studio both = and back arrow R proper assignment operators.

(Refer Slide Time: 01:26)

The slide title is "Hierarchy of operations". Below it is the R expression $A = 7 - 2 * 3^2 + 4$. To the left of the expression is a table:

Order of Precedence	Operation
<i>Bracket</i>	()
<i>Exponent</i>	${}^{\wedge} \text{**}$
<i>Division</i>	/
<i>Multiplication</i>	*
<i>Addition and subtraction</i>	+,-

Below the table is the footer "Arithmetic, Logical and Matrix operations". On the right side of the slide, there is a small video thumbnail of a man speaking.

Let us look at the hierarchy of operations while performing the arithmetic operations in R. So, it is similar to our normal BODMAS rule with bracket has the first importance exponent has the second priority and followed by division, multiplication, addition and subtraction. For your understanding you can type in this expression and then see what is the value of a would be if you want to understand the order of precedence first we do not have any brackets in here.

The next one is exponent the first this part 3^2 will be evaluated that is 9 and the next operation is division $27 / 9$ will give you 3, 3 times 2 is 6 because the next operation is multiplication. So, once you have 6 here what is the next operation? Addition 6 is - 6 because you have - 1 here $7 + 4$ is 11, - 6 and which gives you value of A as 5.

(Refer Slide Time: 02:35)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header is a large white area with a rounded rectangular callout containing the title 'Logical operations in R'. In the bottom right corner of the slide area, there is a small video frame showing a man in a blue shirt speaking. At the very bottom of the slide, there is a thin dark bar with the text 'Arithmetic, Logical and Matrix operations'.

Next we move on to the logical operations in R. So, we have standard logical operations such as $<$, \leq , $>$, \geq , $=$ and so on.

(Refer Slide Time: 02:39)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header is a large white area with a rounded rectangular callout containing the title 'Logical operations in R'. To the right of the title, there is a table with three columns: 'Symbols', 'Operation', and 'Examples'. The table lists the following entries:

Symbols	Operation	Examples
$<$	Less than	<code>> 2<3 [1] FALSE</code>
\leq	Less than equal to	<code>> 2<=3 [1] TRUE</code>
$>$	Greater than	<code>> 2>3 [1] FALSE</code>
\geq	Greater than equal to	<code>> 2>=3 [1] FALSE</code>
$=$	Exactly equal to	<code>> 2==3 [1] TRUE</code>
\neq	Not equal to	<code>> 2!=3 [1] FALSE</code>
!	Not	<code>> !2 [1] FALSE</code>
$ $	Or	<code>> 2 3 [1] TRUE</code>
$\&$	And	<code>> 1&1 [1] 1</code>
isTRUE	Test if variable is TRUE	

At the bottom of the slide, there is a thin dark bar with the text 'Arithmetic, Logical and Matrix operations'.

There are examples where you can see if you ask 2 is greater than 3 it will true a value false because this statement to greater than 3 is not true. Similarly if you say 2 = 3 it will also say false because 2 \neq 3. When you execute this command 2 \neq 3 it will give answer as true because 2 \neq 3. So, this is the summary of logical operations that can be performed in R.

(Refer Slide Time: 03:21)

The slide has a dark blue header bar with the text 'Data science for Engineers'. Below the header is a large white area containing the title 'Matrix operations in R' in a bold, black font, enclosed in a rounded rectangular border. In the bottom right corner of the white area, there is a small video thumbnail showing a man in a blue shirt speaking.

Next we move to the important class of operations that are needed for data analysis problems. Most of the data we will treat them as matrices. So, matrix operations play a key R important role by solving the data analysis problems.

(Refer Slide Time: 03:38)

The slide has a dark blue header bar with the text 'Data science for Engineers'. Below the header is a white area containing the title 'Matrices' in a large, bold, black font. A blue callout box contains the text: 'A matrix is a rectangular arrangement of numbers in rows and columns' and 'Rows run horizontally and columns run vertically'. Below the callout box are two examples of matrices. The first is a 3x3 matrix with elements 1, 5, 3; 4, 9, 2; and 5, 6, 7. The second is a 3x1 column vector with elements 1, 2, 3. In the bottom right corner of the white area, there is a small video thumbnail showing a man in a blue shirt speaking.

Let us first define what matrices are. A matrix is a rectangular arrangement of numbers in rows and columns in a matrix as we know rows are the ones which run horizontally and columns are the ones which run vertically. These are the examples of matrices. This

matrix has 3 rows and 3 columns, and this matrix has 3 rows and 1 column, and this has 1 row and 3 columns.

(Refer Slide Time: 04:06)

Creating matrices

Follow these steps to create a matrix

1. Open a curly bracket,
A = matrix()
2. Enter the sequence of elements,
A = matrix(c(1,2,3,4,5,6,7,8,9))
3. Specify the parameters nrow, ncol, byrow
A = matrix(c(1,2,3,4,5,6,7,8,9), nrow = 3, ncol=3, byrow=TRUE)

This parameter decides how values in the vector would be assigned i.e. "by row" or not

```
> A
> 
> A
[1] [2] [3]
[1,] 1 2 3
[2,] 4 5 6
[3,] 7 8 9
```

Now, let us see how to create matrices in R. To create a matrix in R you need to use the function called `matrix`. The arguments to this matrix are the set of elements that are needed to be the elements of the matrix. You have to pass how many number of rows, you want to have how many number of columns, you want to have in your matrix and this is the important one by row usually R arranges the elements you have entered in a column fashion, if you want the elements that are given to be entered in a row as fashion you have to say by row as true the default option for by row is false.

Now, we have seen; what are the things that are involved in creating a matrix. Let us create a matrix with the elements 1 to 9 which is containing 3 rows and 3 columns and you want to fill the elements in a row wise fashion this is the command which does this and if you see the output is 1 2 3 4 5 6 7 8 9 that are filled in a row wise fashion.

(Refer Slide Time: 05:10)

Different ways of creating matrices:

- Matrix where all rows and columns are filled by a single constant 'k'.
 - For k=3, with 'm' rows & 'n' columns
 - Command :matrix(3,m,n)
- Diagonal matrix:
 - Values in diagonal, similar to 'matrix()'.
 - Mention 'k' as constant/array in first parameter.
 - Command: diag(k,m,n)
- Identity matrix:
 - Use 'diag()' command with k=1

```
Console > matrix(3,3,4)
> [,1] [,2] [,3] [,4]
[1,] 3 3 3 3
[2,] 3 3 3 3
[3,] 3 3 3 3
> diag(c(4,5,6),3,3)
> [,1] [,2] [,3]
[1,] 4 0 0
[2,] 0 5 0
[3,] 0 0 6
> diag(1,3,3)
> [,1] [,2] [,3]
[1,] 1 0 0
[2,] 0 1 0
[3,] 0 0 1
```

Now, let us see how to create some fashion matrices in R the first one is scalar matrix which contains all the rows and columns that are filled by single constant k. So, we need to specify the value to be 3 and you have to specify the number of rows you want and the number of columns you want. So, you want to fill all the rows and columns with the element 3 which is a matrix which contains 3 rows and 4 columns. So, you have specified 3, 3 and 4 when you do that you will get the matrix printed like this.

So, the command is matrix this is the element you want to print in all the rows and columns you have to specify how many rows and how many columns. Next we see how to create diagonal matrix the inputs you have to give for the diagonal matrix is the elements which you want to have in the diagonal and the dimension of the matrix. So, this is the command diag, the elements are vector of elements you want to have as diagonal elements and the rows and number of columns. So, see this example we want 4 5 6 ask the elements of our diagonals and you want to have A₃ by 3 matrix you can use this command and you can see that 4 5 and 6 are your elements in the diagonal and the rest of the elements are there.

How do you create identity matrix? You can create an identity matrices in the diag command with the values in the diagonals has to be 1 and then let us say you want to create A 3 by 3 identity matrix you have to specify then rows as 3 and number of columns as 3 and it will put 1 in the diagonals with all other elements as 0.

(Refer Slide Time: 07:00)

Data science for Engineers

Exercise: Creating matrices

Create the following matrices in R

$$\begin{bmatrix} 3 & 5 \\ -2 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 10 \\ 3 & -1 \\ 7 & 5 \end{bmatrix}$$

and

$$\begin{bmatrix} 2 & 3 & 4 \\ 0 & 1 & 2 \\ -1 & -2 & -3 \\ 5 & 4 & 3 \end{bmatrix}$$


Now, as an exercise you can try creating the following matrices in R.

(Refer Slide Time: 07:04)

Data science for Engineers

Matrix metrics

```
# create a matrix A
A<-matrix(c(1,2,3,4,5,6,7,8,9), nrow =3,
ncol=3, byrow=TRUE)

Finding the size of the matrix, A :
dim(A) will return the size of the matrix
nrow(A) will return the number of rows
ncol(A) will return the number of columns
prod(dim(A)) or length(A) will return the number of elements
```

Console ~ / ↘

```
> dim(A)
[1] 3 3
> nrow(A)
[1] 3
> ncol(A)
[1] 3
> length(A)
[1] 9 .
> |
```



Next we move on to matrix metrics once a matrix is created how can you know the dimension of the matrix? How can you know how many rows are there in the matrix? How many columns are in the matrix? How many elements are there in the matrix is the questions we generally wanted to answer.

We can use the following comments to know all of this. Dimension of A will return the size of the matrix that will say what is the size of the matrix that is it is A 3 by 3 or 4 by

5 and so on, n row of a will return you number of rows and n column of you will return you number of columns. Either length of a or product of dimensions of A will return the number of elements that are existing in the matrix. For the matrix A which is created by using this command we can find that dimension of A will give you 3 by 3 because it contains 3 rows and 3 columns number of rows is 3 and number of columns is 3 and the number of elements that are present in the matrix is 9.

(Refer Slide Time: 08:09)

The slide is titled "Accessing, editing, deleting in elements in matrices". It includes the following text and code:

They follow the same convention as dataframes such as

- Array/value before "," for accessing rows
- Array/value before ":" for accessing columns
- use of "-" for removing rows/columns
- Strings can be assigned as names of rows and columns using:
 - `rownames()` and `Colnames()`

```

> A=matrix(c(1,2,3,4,5,6,8,9,1),
+           3,3,byrow = T)
>
> colnames(A) <- c("a","b","c")
> rownames(A) <- c("d","e","f")
> A
   a b c
d 1 2 3
e 4 5 6
f 8 9 1
> A[,1:2]
   a b
d 1 2
e 4 5
f 8 9
> A[,c("a","c")]
   a c
d 1 3
e 4 6
f 8 1
> A[-c("d","f"),]
   a b c
d 1 2 3
e 8 9 1
> |

```

Arithmetic, Logical and Matrix operations

We can access, edit and delete elements in the matrices using the same convention that is followed in data frames. So, you will have a matrix and followed by a square bracket with a comma in between array and values before the comma is used to access rows and array or value that is after comma is used to access columns. If you want to remove some columns you need to add a negative symbol before the rows or columns, and you can also assign strings as names of rows and columns by using the commands row names and row columns.

Here we have created a matrix A which are having the elements 1 2 3 4 5 6 8 9 1 and it is a 3 by 3 matrix and we want to fill the elements row wise and we can now name the columns as a b c and name the rows as d e f. Once you do that and print a you can see that this column is named as a, and this column is named as b, and this column is named as c. Similarly we can see that row one is named as d, row 2 is named as e and row 3 is named as f.

Now, let us suppose you want to access the first two columns you can use the same convention as what we have used for data frames, A with the square bracket nothing before the comma and then you want access 1 to 2 that is first two columns of a you have to give that array here and then it will access the first two columns of A.

You can also access the columns using the names of the column as we have seen in the data frames. So, you want to access the columns a and c; that means, columns 1 and 3 you can do so, by specifying the names of the columns. Similarly you can also access the rows by using the names of the rows. You want to access first and third row which are having the names d and f, you can do so by using this command you want access row d and row f and all the columns. So, the output is shown here.

(Refer Slide Time: 10:30)

Data science for Engineers

Accessing an entry of a matrix

$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

First row, second column
 $\rightarrow A[1,2]$
 $[1] 2$

Second row, third column
 $\rightarrow A[2,3]$
 $[1] 6$

The part before the comma should be the row number
The part after the comma should be the column number

Arithmetic, Logical and Matrix operations

If you want to access an entry of a matrix you can use the similar convention. For example if you want to access this element it is in the first row and the second column the command you need to use is in the matrix A fetch the element which is in the first row and in the second column that will give you the output 2. And for example, if you want to access this element 6 you have to say it is in the second row and the third column you have to say A of 2 comma 3, it is give an output 6. As we have seen earlier the part before the comma should refer to the row number and the part after the comma should refer to the column number.

(Refer Slide Time: 11:12)

Data science for Engineers

Accessing a column

- Specify the column index
- Leave the rows index unspecified
- This means accessing all row elements of the given column index

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

> A[,1]

[1] 1 4 7

>

All rows in first column

Arithmetic, Logical and Matrix operations

16

Now, let us see how to access a column of a matrix. So, specify the column index which you want access and leave the rows index unspecified. This means you are accessing all the row elements of a given column index. So, for example, if you want to access first column of the matrix A, what you need to do is A of all the rows and first column which will give you the output 1 4 7.

(Refer Slide Time: 11:41)

Data science for Engineers

Accessing a row

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

> A[2,]

[1] 4 5 6

>

Leaving the column index empty means choose all the columns

How do you access the last row ?
A[nrow(A),]

Arithmetic, Logical and Matrix operations

17

Similar to accessing a column we can access a row of a matrix. What you need to do is you need to specify the row which you want to access and specify nothing in the column

index which says access all the columns. If you want to access row 2 you have to specify in the row ID as 2 and leave empty space in the column ID and so that row two all the columns will print it and you will be able to access 4 5 6.

For you to think about how do you access the last row. Can you do something like this? You figure out by trying on your own.

(Refer Slide Time: 12:20)

The screenshot shows a MATLAB-like interface. At the top, a dark bar reads "Data science for Engineers". Below it, a light gray bar displays the title "Accessing everything but one column". The main workspace shows a 3x3 matrix A:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

A red box highlights the second column (2, 5, 8). Below the matrix, a list of bullet points explains how to access this column:

- Access the column that has to be avoided and then put a '-' sign in front of it
- For example: `A[:, -2]`
- This will fetch all the columns except the 2nd column

Below the list, a command prompt shows the code `> A[:, -2]` followed by the resulting output:

```
> A[:, -2]
[1,]    1    3
[2,]    4    6
[3,]    7    9
```

On the right side of the slide, there is a small video thumbnail of a person speaking.

Next we will see how do access everything, but one column. I want to access in this matrix this part 1 4 7 and 3 6 9 I do not want this column to be in the matrix where I want to access.

So, now what I have to do is it is like eliminating this column from the matrix you can do so by having a negative symbol before this is the second column you can say all the rows I want and I want to take this second column off and if I assign it back to A, I will get A as 1 4 7 and 3 6 9 or if you just print this a of all comma - 2 it will give the desired result which is 1 4 7 and 3 6 9.

(Refer Slide Time: 13:09)

Data science for Engineers

Accessing everything but one row

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```
> A[-2,]
[1,] 1 2 3
[2,] 7 8 9
```

- Access the row that has to be avoided and then put a '-' sign in front of it
 - For example: `A[-2,]`
 - This will fetch all the row except the 2nd row



Arithmetic, Logical and Matrix operations

Similar to the one which you have seen in the earlier slide you can also access everything, but one row all you need to do is for example, if you want to access all the parts of a except this row you can do so by using this command I want to take the second row off and I want to have all the columns. Now, once when you do this command you will say 1 2 3 and 7 8 9 will be printed as your output.

(Refer Slide Time: 13:40)

Data science for Engineers

Exercise: Accessing elements of a matrix

Do the following in R

Assign the following matrix

$$A = \begin{bmatrix} 1 & 7 & 3 \\ 4 & 4 & 6 \\ 4 & 7 & 12 \end{bmatrix}$$

- Change the element 12 to 13
- Access the second row and the third column
- List all the elements in the second column and third row



Arithmetic, Logical and Matrix operations

As an exercise to access elements of a matrix you can try solving this problems that are given.

(Refer Slide Time: 13:46)

Data science for Engineers

Colon operator

Colon operator can be used to create a row matrix

```
> 1:10  
[1] 1 2 3 4 5 6 7 8 9 10  
>  
> 10:1  
[1] 10 9 8 7 6 5 4 3 2 1  
>
```



Arithmetic, Logical and Matrix operations

Now, we will introduce what is called as a colon operator. Colon operator is used to create an array of elements with equal width for example, if I type in 1 to 10 it will create numbers from 1 to 10 with gap of 1. I can also reverse the order it will print from 10 to 1 with a gap of 1. Why is this colon important? If you would have realized I would have used something similar while accessing the number of rows or columns in the previous slides. Let us look how to do this.

(Refer Slide Time: 14:22)

Data science for Engineers

Colon operator: sub matrices selection

The colon notation can also be used to pick sub-matrices

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}_{3 \times 3}$$

Sub-matrix

The sub-matrix occupies the **first three rows** and the **first two columns**

```
> A[1:3,1:2]      > A[1:3,-3]      > A[,1:2]  
[1,] [1] 2      [1,] 1 2      [1,] 1 2  
[2,] 4 5      [2,] 4 5      [2,] 4 5  
[3,] 7 8      [3,] 7 8      [3,] 7 8  
>                 >                 >
```



Arithmetic, Logical and Matrix operations

For example if you want to select a part of matrix which has sub matrix you can use this colon operator ok. So, let us now see if I want to access the first 3 rows and the first 2 columns of this matrix, how do I do this? I want to access rows 1 to 3 and also access columns 1 to 2 do. So, you can see this colon operator is helping us in accessing the sub matrices from the matrix.

In this example what does it says is I want to access all the 3 rows and I do not want the third column. This is same operation, but done in a different fashion. You can also do the same I want to access all the rows, but it has to be coming from first two columns only. So, you can see that you can access sub matrix in different fashions depending upon the way you are comfortable with.

(Refer Slide Time: 15:20)

The image shows a MATLAB interface. At the top, a dark bar displays "Data science for Engineers". Below it, a title bar says "Accessing submatrices: Example 2". The main workspace shows a 3x3 matrix A and its 2x2 submatrix. The matrix A is defined as:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Its submatrix A[1:3, 1:2] is extracted and displayed as:

$$\begin{bmatrix} 1 & 2 \\ 7 & 8 \end{bmatrix}$$

Below the matrix, two command-line entries are shown in a blue-bordered box:

```
> A[c(1,3),1:2]
> A[c(1,3),c(1,2)]
```

On the right side of the slide, there is a small video frame showing a person speaking.

So, this is another example of accessing sub matrices if I want to access this 1 comma 2 and 7 comma 8 and have it as a sub matrix separately how do I do this. I want to access rows 1 and 3 and what are the columns I need to access in the columns 1 and 2. So, I have to say in the columns 1 and 2 access the elements which are in the row 1 and row 3, that brings me the matrix. You can use the concatenation operator also for both the arguments like shown here you can use c of 1 comma 3 and c of 1 comma 2 which gives you the desired result.

(Refer Slide Time: 16:00)

Data science for Engineers

Exercise: Accessing sub-matrices

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

How do you access this sub-matrix

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



Arithmetic, Logical and Matrix operations

You can try this as an exercise for accessing sub matrices.

(Refer Slide Time: 16:06)

Data science for Engineers

Matrix concatenation

- Matrix concatenation refers to merging of a row or column to a matrix
- Concatenation of a row to a matrix is done using `rbind()`
- Concatenation of a column to a matrix is done using `cbind()`
- Consistency of the dimensions between the matrix and the vector should be checked before concatenation



Arithmetic, Logical and Matrix operations

Next we move on to another important operation on matrices which is matrix concatenation. Matrix concatenation refers to merging of rows or columns to an existing matrix. If you want to add a row to the existing matrix you can do so by using R bind command. If you want to add a column to a matrix you can do so by using c bind command. So, one thing you have to keep in mind is you have to make sure the

consistency of dimensions before you do this matrix concatenation. Let us illustrate how an R bind works.

(Refer Slide Time: 16:38)

Data science for Engineers

Matrix concatenation – rbind()

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad B = \begin{bmatrix} 10 & 11 & 12 \end{bmatrix}$$

Use rbind() to append B row vector to the rows of A

$$C = \begin{bmatrix} A \\ B \end{bmatrix}$$

```
> C = rbind(A,B)
>
> C
 [,1] [,2] [,3]
 [1,] 1 2 3
 [2,] 4 5 6
 [3,] 7 8 9
 [4,] 10 11 12
```

Arithmetic, Logical and Matrix operations

Let us suppose we have a matrix A and matrix B and you want to concatenate this matrix B as a row in matrix A that can be done using the R bind command which is shown here. I am concatenating matrix B to the matrix A and I am assigning it to the variable C. So, when you do this command you can see that the matrix C is having the row 10 11 12 which is the matrix B and is concatenated to the matrix A.

(Refer Slide Time: 17:13)

Data science for Engineers

Matrix concatenation – cbind()

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad B = \begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

Use cbind() to append B column vector to the columns of A

$$C = \begin{bmatrix} A & B \end{bmatrix}$$

```
> C = cbind(A,B)
>
> C
 [,1] [,2] [,3] [,4]
 [1,] 1 2 3 10
 [2,] 4 5 6 11
 [3,] 7 8 9 12
```

Arithmetic, Logical and Matrix operations

Now, let us see the C bind. Let us say you have this matrix A and we have matrix B which is shown in the screen you want to concatenate this B matrix with the columns of A. You can do so by using the C bind command which is shown here C by pass the first matrix A and second matrix B and assign it to the variable C. When you print the C you can see that the matrix B has been concatenated as a column to the matrix A.

(Refer Slide Time: 17:50)

Data science for Engineers

Dimension inconsistency –cbind()

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad B = [10 \quad 11 \quad 12]$$

Can these two matrices be merged to give

$$C = [A \quad B]$$

```
> D = cbind(A,B)
Error in cbind(A, B) : number of rows of matrices must match (see arg 2)
```

Arithmetic, Logical and Matrix operations



Now, let us try to concatenate this B to this matrix A using C bind. What would do you expect? We expect an error because A is having the dimension 3 by 3, but B is having 1 by 3. If I want to do a column bind the dimension of matrix B would have been 3 by 1, but it is 1 by 3 which is inconsistent that is why you will get an error, error in C bind of A number of matrices must match.

(Refer Slide Time: 18:26)

Data science for Engineers

Fixing the dimension inconsistency

```
A = 
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

B = 
$$\begin{bmatrix} 10 \\ 11 \\ 12 \end{bmatrix}$$

C = 
$$\begin{bmatrix} A & B \end{bmatrix}$$

```

```
> C = cbind(A,B)
>
> C
     [,1] [,2] [,3] [,4]
[1,]    1    2    3   10
[2,]    4    5    6   11
[3,]    7    8    9   12
> |
```

Arithmetic, Logical and Matrix operations

Now, if you want to resolve this dimension inconsistency you have to transpose this B and then have this as 3 by 1 and now A is 3 by 1 now you can easily do the C bind operation by using C bind command C bind of A comma B and assign it to C. Now, you can see that this C bind it happened and the B is concatenated to the matrix A.

(Refer Slide Time: 18:52)

Data science for Engineers

Deleting a column

```
A = 
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

```

- Access the column that has to be deleted and then put a '-' sign in front of it
 - For example: $A=A[, -2]$
 - This will fetch all the columns except the 2nd column

```
> A[, -2]
     [,1] [,2]
[1,]    1    3
[2,]    4    6
[3,]    7    9
```

Arithmetic, Logical and Matrix operations

You have seen how to delete a column, you can use negative symbol before the columns which you want to delete and then assign it to A you will see that the required output is printed.

(Refer Slide Time: 19:05)

Data science for Engineers

Deleting a row

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- Access the row that has to be deleted and then put a '-' sign in front of it
 - For example: $A=A[-2,]$
 - This will fetch all the rows except the 2nd row

```
> A[-2, ]
[,1] [,2] [,3]
[1,] 1 2 3
[2,] 7 8 9
>
```

Arithmetic, Logical and Matrix operations

Similar to what we have seen in the earlier slide we can also delete a row from the matrix which is, let us suppose we want to delete this row 2 you have to say - 2 and then all columns and then assign it back to A. You can see that in the output the row 2 is deleted.

(Refer Slide Time: 19:23)

Data science for Engineers

Matrix algebra

- Addition/subtraction
- Multiplication
- Matrix Operations in R
- Matrix Division

Arithmetic, Logical and Matrix operations

Now, let us see how to do algebraic operations on matrices such as addition, subtraction, multiplication and matrix division in R.

(Refer Slide Time: 19:35)

The screenshot shows a Jupyter Notebook interface with the title "Matrix addition/subtraction & multiplication". It displays three code cells and their outputs:

- Cell 1: $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 8 & 9 & 1 \end{bmatrix}_{3 \times 3}$
- Cell 2: $B = \begin{bmatrix} 3 & 1 & 3 \\ 4 & 2 & 1 \\ 5 & 1 & 2 \end{bmatrix}_{3 \times 3}$
- Cell 3: Shows the output of $A + B$, which is a matrix where each element is the sum of the corresponding elements from matrices A and B.
- Cell 4: Shows the output of $A * B$ (regular matrix multiplication), resulting in a matrix where each element is the product of the corresponding elements from matrices A and B.
- Cell 5: Shows the output of $A.*B$ (element-wise multiplication), resulting in a matrix where each element is the element-wise product of the corresponding elements from matrices A and B.

A callout box highlights the output of Cell 5, labeled "Element-wise multiplication is based on multiplication between corresponding elements of two matrices." A red arrow points from this text to the highlighted output.

Let us suppose we have two matrices A and B which are shown here. Matrix addition is straight forward you can say $A + B$ you will get the output. So, $1 + 3$ is 4 , $2 + 1$ is 3 , and $3 + 3$ is 6 you will see the element wise operation happens that is what normal matrix operation is also about.

So, you can also do the subtraction, multiplication is little bit trickier when you say A has trick B it will perform element wise multiplication such as 1 into 3 is 3 , 2 into 1 is 2 and 3 into 3 is 9 . But if you want to have a regular matrix multiplication you have to use percentage symbol before and after this hash trick that will perform the regular matrix operation.

(Refer Slide Time: 20:26)

The screenshot shows a RStudio interface. At the top, it says "Data science for Engineers". Below that is a title "Matrix division". A red warning icon with the word "WARNING:" is present. The text inside the warning box reads: "The following operation is not inverse of a matrix but element wise division between matrices A & B." To the right of the warning box is a "Console" window showing R code and its output. The code defines two matrices, A and B, and then performs an element-wise division of A by B. The formula $A/B = \frac{a_{ij}}{b_{ij}}$ is displayed. The matrices are:

$$A = \begin{bmatrix} 4 & 9 \\ 16 & 25 \end{bmatrix}, B = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

Console output:

```
> A<-matrix(c(4,9,16,25),2,2)
> B<-matrix(c(2,3,4,5),2,2)
> A
[,1] [,2]
[1,]    4    16
[2,]    9    25
> B
[,1] [,2]
[1,]    2    4
[2,]    3    5
> A/B
[,1] [,2]
[1,]    2    4
[2,]    3    5
```

At the bottom of the RStudio interface, there is a bar with the text "Arithmetic, Logical and Matrix operations". On the right side of the slide, there is a small video thumbnail of a man speaking.

Now, let us look at matrix division. Let us say you have two matrices A and B which are 4 9 16 25, and 2 3 4 5 respectively. Now, if I do A by B what it does is element wise division, but not the inverse of a matrix. So, you have created matrix A matrix B and then if you do A by B you will see that 4 by 2 is 2 9 by 3 is 3, 16 by 4 is 4. So, let us suppose you have two matrices A and B as shown in the figure when you do A by B it will perform an element wise division, but not the inverse of a matrix.

In this video we have seen how to do arithmetic logical and matrix operations in R. In the next lecture we are going to discuss about how to write functions in R, and how to invoke them, how to use them to perform the task we wanted.

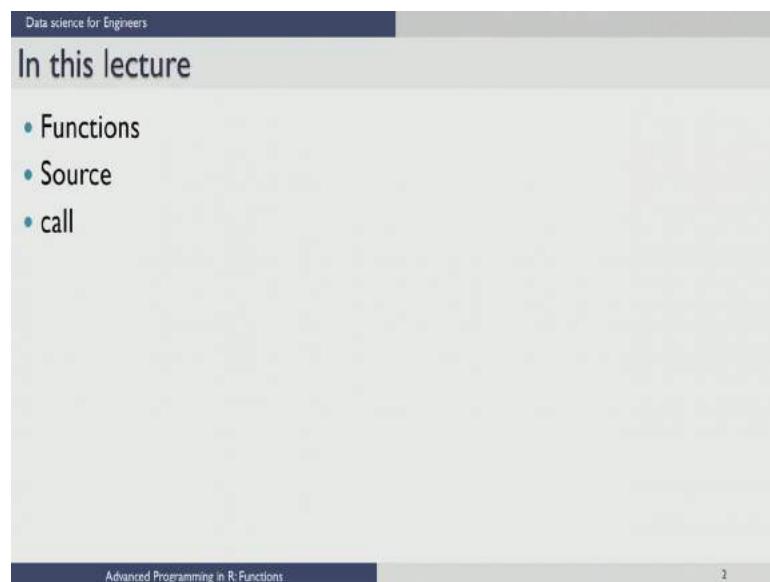
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 08
Advanced programming in R: Functions

Welcome to the lecture 7 in the R module of the course Data Science for Engineers.

(Refer Slide Time: 00:21)



In this lecture we are going to introduce you to the functions in R. We are going to explain how to load or source the functions and how to call or invoke the functions.

(Refer Slide Time: 00:32)

The screenshot shows a presentation slide titled "Data science for Engineers" at the top left. The main title of the slide is "Functions in R". Below the title is a bulleted list of four points:

- A function accepts input arguments and produces output by executing valid R commands present in the function.
- Function name and file names need not be the same.
- A file can have one or more function definitions.
- Functions are created using the command `function()`

Below the list is a code snippet in a box:

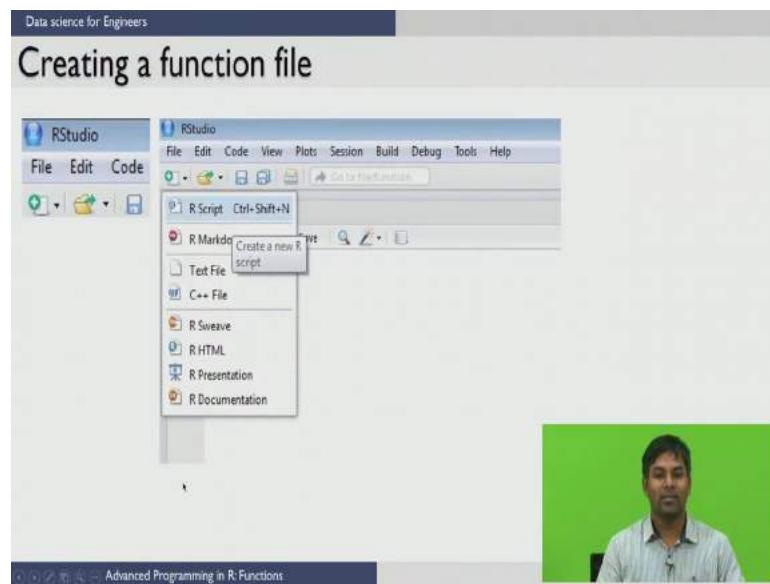
```
f = function(arguments) {  
    statements  
}
```

A small video thumbnail of a man speaking is visible on the right side of the slide.

Functions are useful when you want to perform certain tasks many number of times. A function accepts input arguments and produces the output by executing valid R commands that are inside the function.

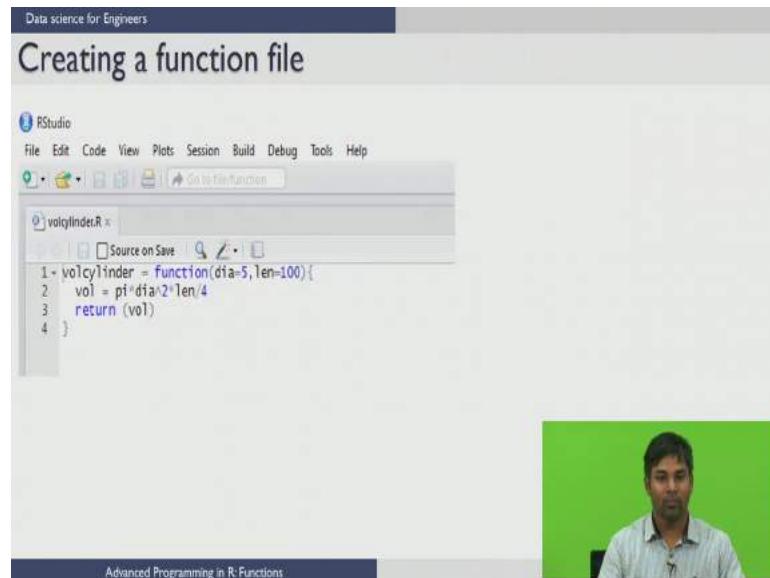
In R when you are creating a function the function name and the file in which you are creating the function need not be same and you can have one or more function definitions in a single R file. Functions are created in R by using the command `function`. The general structure of the function file is as follows `f = function of arguments and then you have statements that are needed to be executed`. This `f` is the function name when you write this command this means that you are creating a function with name `f` which takes certain arguments and executes the following statements.

(Refer Slide Time: 01:31)



Let us see how to create a function file. Creating a function file is similar to opening an R script which we have already seen. You can either use file button in the toolbar or you can use the + button just below the file tab to create an R script, once you create an R script you can save it with whatever name you want.

(Refer Slide Time: 01:56)

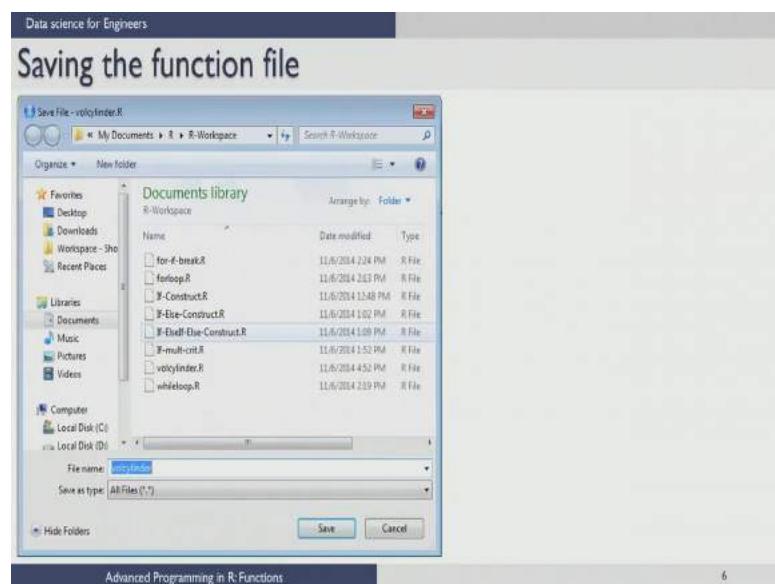


For example, we have saved the R file as vol cylinder. Now, once you save you are ready to write functions, now I want to create a function which calculates the volume of a cylinder which takes in the arguments the diameter and length. So, to create a function

by name volume cylinder I have to have the function named as volume cylinder function and the arguments that are needed to be passed are the diameter of the cylinder and the length of the cylinder.

If you notice here we are passing this values of 5 and 100 as a default arguments for this function. Once you have diameter and length you can calculate the volume by the formula $\pi d^2 l / 4$ then what you need to return is the volume variable that is calculated inside the function. Once you have written the R statements that are needed to be executed in a function file, you can save that file. So, we are saving it as vol cylinder dot R.

(Refer Slide Time: 02:56)



Once you save this. So, you need to load the functions before you invoke or execute them in R. To load a function you need to click on the source button that is available in the R script menu.

(Refer Slide Time: 03:02)

The slide has a dark blue header bar with the text 'Data science for Engineers'. The main title 'Loading the functions' is in bold black font. Below it, a callout box contains the text 'Function files have to be loaded before invoking (execution)'. A sub-section title 'Loading a function file' is followed by a screenshot of a software interface showing a toolbar with 'Run', 'Source' (highlighted in blue), and other buttons. A note below the interface states: 'The function file can also be loaded using the following command > source('~/R/R-Workspace/volcylinder.R')'. Another note in a box says: 'Note: Clicking the "Source" button will not execute the function, it will only load the function file. After loading, the function can be executed by invoking the function'. At the bottom, there is a footer bar with the text 'Advanced Programming in R Functions' and a small number '7'.

Clicking the source button will not execute the function; it will only load the function file and make it ready for invoking.

(Refer Slide Time: 03:26)

The slide has a dark blue header bar with the text 'Data science for Engineers'. The main title 'Invoking the function from console' is in bold black font. Below it, a code block shows R commands: > source('~/R/R-Workspace/volcylinder.R') > v = volcylinder(5,10) > v [1] 196.3495. To the right, a 'Variable Browser' window is shown with tabs 'Environment' and 'History'. It lists variables 'v' with value '196.349540849362' and a function 'volcylinder' with definition 'function (dia, len)'. At the bottom, there is a footer bar with the text 'Advanced Programming in R Functions' and a small number '8'.

Once you load the function, you can invoke the function from the console as follows you want the volume to be saved in the variable v and then you are calling this function vol cylinder with the arguments 5 and 10. So, this will run the function to calculate the volume and returns the volume. In the variable browser you can also see value of volume

and you can also see that the function volume cylinder is available with two arguments dia and length.

(Refer Slide Time: 03:56)

The slide has a dark blue header with the text 'Data science for Engineers'. The main title 'Passing arguments to functions' is in bold black font. Below it is a sub-section title 'Passing variables as arguments to functions'. A bulleted list follows:

- Passed in the same order as in function definition
- Names of the arguments can be used to pass their values in any order
- Default values are used if some or all arguments are not passed

Below the list are three code snippets in a light blue box:

> vol = volcylinder(5,10) > vol [1] 196.3495	> vol = volcylinder() > vol [1] 1963.495
> vol = volcylinder(len = 10, dia = 5) > vol [1] 196.3495	

At the bottom of the slide, there is a navigation bar with icons for back, forward, search, and other presentation controls. The footer says 'Advanced Programming in R: Functions' and has a page number '9'.

Now, there are several ways you can pass the arguments to the function. Generally in R the arguments are passed to the function in the same order as in the function definition. If you do not want to follow any order what you can do is you can pass the arguments using the names of the arguments in any order. If the arguments are not passed the default values are used to execute the function.

Now, let us see the examples for each of these cases when you pass the arguments 5 and 10 the first argument is diameter and second argument is length according to the definition of the function. So, it will take in the same way, but when you want not to follow any order you can pass the arguments by the names in any order. So, for example, I want to pass length as a first argument you can specify length = 10 and diameter = 5 and you can still see the result is same even though you pass the arguments in the different order.

So, point to keep in mind is you can pass the arguments in any order by specifying its name. When you do not pass any arguments here it takes the default values of 5 and 100 which are default diameter and length and then calculates the volume.

(Refer Slide Time: 05:26)

Data science for Engineers

Lazy evaluations of functions in R

- Functions are lazily evaluated, which means that if some arguments are missing, the function is still executed as long as the execution doesn't involve these arguments

```
> volcylinder = function(dia, len, rad){  
+ vol = pi*dia^2*len/4  
+ return(vol)}  
>  
> vol = volcylinder(dia = 5, len = 10)  
> vol  
[1] 196.3495
```

Argument `rad` is missing, but the function is executed

```
> volcylinder = function(dia, len, rad){  
+ vol = pi*dia^2*len/4  
+ print(rad),  
+ return(vol)}  
>  
> vol = volcylinder(dia = 5, len = 10)  
Error in print(rad) : argument "rad" is missing, with no default
```

Here `rad` is used in the function body, which throws up error

Advanced Programming in R Functions

A video player shows a man speaking.

In R the functions are executed in a lazy fashion, when we say lazy what it mean is if some arguments are missing the function is still executed as long as the execution does not involve those arguments. We will see an example for this. We have the same function we have defined now an extra argument radius in the function and the volume calculation does not involve this argument radius in this calculation.

Now, when you pass this arguments dia and length even though you are not passing this radius the function will still execute because this radius is not used in the calculations inside the function. But R is clever enough, if you do not pass the argument and then use it in the definition of the function it will throw an error saying that this rad is not passed and it is being used in the function definition.

(Refer Slide Time: 06:28)

Summary of function file creation and execution

1. Open a function file by clicking . First line of a function file should be `function_name = function (inputs)`. Type the necessary and valid R statements/commands to be executed
2. Save the function file
3. Load the function file by pressing
4. Invoke the function with the right number of inputs to execute the function

In summary these are the steps in creating a function file in R and executing. First we need to open or create a function file by clicking a that the + symbol or file tab in the toolbar you have to define the function in this fashion function name, keyword function and the input arguments.

All the statements that are typed inside the function has to be valid R statements to be executed, and you need to save the function file before executing you need to load the function file by using the source button once you load the function file you can invoke or call the function file with the right number of inputs so that you will execute the function properly and you will get the required result.

(Refer Slide Time: 07:23)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. The main content area has a light gray background. At the top left, the title 'Final word' is displayed in bold black font. Below the title, there is a block of text in black font: 'Have to load the function file every time when you clear the console, restart R or make changes in the function file'. Underneath this text, there is a code snippet in a monospaced font. The first line is 'gt; volcylinder(5,10)', and the second line is 'Error: could not find function "volcylinder"'. At the bottom of the slide, there is a dark blue footer bar with the text 'Advanced Programming in R Functions' on the left and '(1)' on the right.

A final word we need to load the function file every time you change something inside the function definition either you restart R studio or make changes in the function file. If you do not do that either you get an error or you will not get correct outputs which you are expecting, because you would have changed something in the function definition and not saved the original version. Once you save the original version also you have to invoke the function before you use.

In the next lecture we are going to explain the functions which are having multiple inputs and multiple outputs.

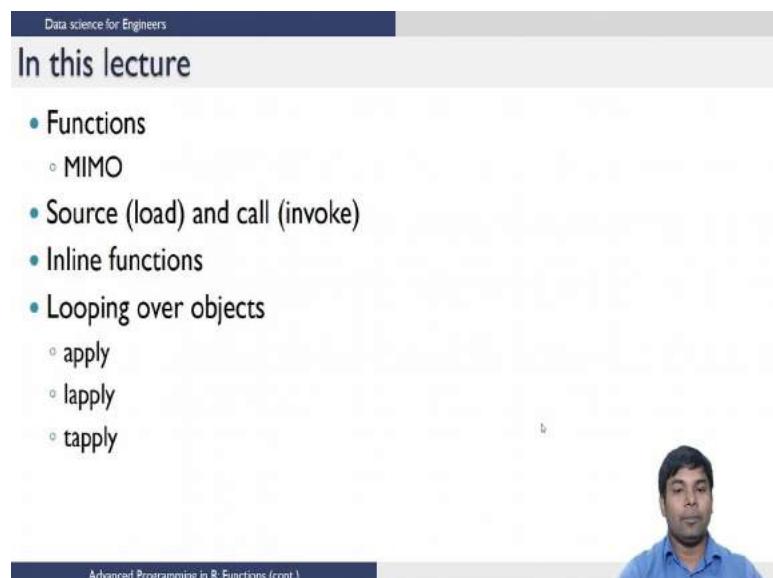
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 09
Advanced programming in R: Functions

Welcome to lecture 8 in the R module of the course Data Science for Engineers. In the previous lectures we have seen how to create functions, how to execute them, but we have limited ourselves to the single output.

(Refer Slide Time: 00:33)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header is a light gray section with the title 'In this lecture' in bold black font. Underneath this, there is a white area containing a bulleted list of topics:

- Functions
 - MIMO
- Source (load) and call (invoke)
- Inline functions
- Looping over objects
 - apply
 - lapply
 - tapply

At the bottom of the slide, there is a dark blue footer bar with the text 'Advanced Programming in R: Functions (cont.)' in white.

In this lecture we are going to see functions with multiple inputs and multiple outputs which we call MIMO how to source and call those functions. We will also see about inline functions how to loop over objects using the commands apply, lapply and tapply.

(Refer Slide Time: 00:53)

Data science for Engineers

Multiple input and multiple output functions

Function with multiple inputs and outputs

- Functions in R take multiple input objects but returns only one object as output
- This however is not a limitation, because a list object (collection of several objects) can be returned by function

```
graph LR; DI[Diameter] -->|>| volcylinder_mimo[volcylinder_mimo]; HI[Height] -->|>| volcylinder_mimo; volcylinder_mimo -->|>| Volume[Volume]; volcylinder_mimo -->|>| SA[Surface area]; Volume -->|>| list["list(volume, surface area)"]; SA -->|>| list
```

Advanced Programming in R: Functions (cont.)

Let us see the functions with multiple input and multiple outputs. The functions in R takes multiple input objects, but written only one object as output, this is however, not a limitation because you can create lists of all the outputs which you want to create and once the list is written out you can access the into the elements of the list and get the answers which you want.

Let us consider this example I want to create a function vol cylinder underscore MIMO which takes diameter and height of the cylinder and returns volume and surface area. Since R can written only one object what I have to do is I have to create one object which is a list that contains volume and surface area and return the list. Let us see how we can do that in the next line.

(Refer Slide Time: 01:51)

The screenshot shows the RStudio interface with the title bar "Data science for Engineers" and the main window titled "Creating and saving". A sub-section titled "1. Creating a function file" is displayed. In the RStudio menu bar, "File" is selected, and a dropdown menu is open, showing options like "R Markdown", "R Script", "C/C++ File", "Text File", etc. The "R Script" option is highlighted. Below the menu, a code editor window contains the following R code:

```
1. volcylinder_mimo = function(dia, len){  
2   volume = (pi*dia^2)*len/4 ## Volume in metre^3  
3   surface_area = pi*dia*len  
4   result = list("volume"=volume, "surface_area"=surface_area)  
5   return(result)  
6 }
```

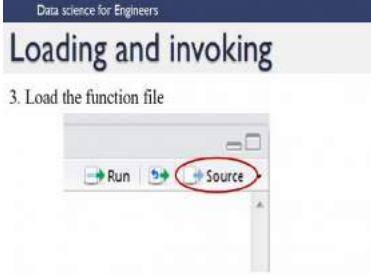
Below the code editor, the status bar shows "4.R" and "volcylinder_mimo.R".

Section 2, "Save the function file "volcylinder_mimo"" is also visible.

So, you need to first create an R file which we have seen several times. You can create an R script using a + button R from the file tab once you have opened R script this is the piece of code that does what we need we want to name the function as vol cylinder underscore MIMO because it is a multiple input and multiple output.

And this function takes an arguments diameter and length earlier we have calculated only volume and now we want to calculate the surface also which is given by π times diameter times length. Since R returns only one object first I need to create an object called result which is a list of volume and surface area, I am naming the volume as volume and surface area surface area I will calculate this result and ask the function to return one object the result which contains both volume and surface area.

(Refer Slide Time: 02:53)



3. Load the function file

4. Execute

```
> source('~/R/R-Workspace/volcylinder_mimo.R')
> result = volcylinder_mimo(10,5)
> result["volume"]
$volume
[1] 392.6991

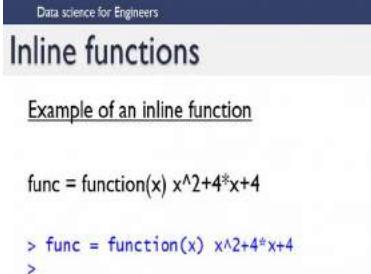
> result["surface_area"]
$surface_area
[1] 157.0796
```

Advanced Programming in R Functions (cont.)



Once you will write the function you need to load the function to call it loading can be done using the source button, once you source it you are ready to call a function. You can call the function once you load the function. So, I am calling this function result = volcylinder_mimo and I am passing 10 and 5 as my arguments this result will contain two elements the first element volume will contain volume the second element surface area will contain surface area. Once the object result is given out by the function I can access inducer elements by using the techniques what we have teached in the list lecture.

(Refer Slide Time: 03:35)



Example of an inline function

```
func = function(x) x^2+4*x+4

> func = function(x) x^2+4*x+4
>
> func(1)
[1] 9
>
> func(2)
[1] 16
>
> func(-2)
[1] 0
>
> func(0)
[1] 4
```

Advanced Programming in R Functions (cont.)



Sometimes creating an R script file loading it, executing it is a lot of work when you want to just create very small functions such as the ones as shown here $x^2 + 4x + 4$. I want to evaluate this expression for different x's. So, having a function file and then loading it, invoking it is a lot of work. So, what we can use in this kind of situations is an inline functions to create an inline function you have to use the function command with the argument x and then the expression of the function.

Once you create this you can call this in the command prompt itself function of one gives takes the value of one as an argument and then x accelerates this expression $1^2 = 1 + 4(1) = 4$, $1 + 4 = 5$, and $5 + 4 = 9$. Similarly you can get the outputs are different arguments which are passed which are shown in the screen.

(Refer Slide Time: 04:42)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. The main content area has a light gray background. At the top left, there is a small circular icon with a play symbol. To its right, the title 'Looping over objects' is displayed in a large, bold, black font. Below the title, there is a bulleted list of text and code snippets:

- There are a few looping functions that are pretty useful when working interactively on a command line
 - **apply**: Apply a function over the margins of an array or matrix
 - **lapply**: Apply a function over a list or a vector
 - **tapply**: Apply a function over a ragged array
 - **mapply**: Multivariate version of lapply
 - **xxply** (plyr package)

At the bottom of the slide, there is a navigation bar with icons for back, forward, and search, followed by the text 'Advanced Programming in R Functions (cont.)'. On the right side of the slide, there is a small video frame showing a person speaking.

So, now, we move on to looping over objects there are few looping functions that are pretty useful when working interactively on a command line few examples are apply, lapply, tapply and so on. Let us see what each of these functions does.

Apply function applies a function over the margins of an array R matrix, lapply function applies a function or a list or a vector, tapply function applies a function over a ragged array and mapply is a multivariate version of lapply. We will see examples for each of this in the coming slides.

(Refer Slide Time: 05:22)

Data science for Engineers

apply function

- Applies a given function over the margins of a given array.
 - Syntax: `apply(array, margins, function,...)`
 - Here margins refer to the dimension of the array along which the function need to be applied.

```
> A = matrix(1:9,3,3)
> A
 [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> apply(A,1,sum)
[1] 12 15 18
> apply(A,2,sum)
[1]  6 15 24
```

Advanced Programming in R: Functions (cont.)

Here is an example for apply function. What apply function does is applies a given function over a margins of a given array, when you say margins here this refers to the dimension of an array along which the function need to be applied.

Let us create a matrix A with the elements 1 to 9 which is of 3 by 3 size. You can do that using this command and when you print A you can see this. Now, I want to evaluate these sums across the rows and the sums across the columns. You can use the apply function to do so, the syntax for this is take the matrix A and then apply this apply function across the rows of matrix A and function you need to apply is sum.

So, now what does is it will sum up this first row $7 + 4 + 1$ it is 12 some of the elements in the second row and prints here, and some of the elements in the third row and prints the output. You can do the same for the columns by specifying the margin as 2 which says, apply the sum function on A across the margin 2 which is the columns. This command will prints the sums of the elements in the columns as $3 + 2 + 1$ is 6 and so on.

(Refer Slide Time: 06:50)

Data science for Engineers

lapply function

- **lapply** is used to apply a function over a list.
- **lapply** always returns a list of the same length as the input list
 - Syntax: `lapply(list, function, ...)`

```
> A=matrix(1:9,3,3)
> B=matrix(10:18,3,3)
> Mylist=list(A,B)
> determinant = lapply(Mylist,det)
> determinant
[[1]]
[1] 0
[[2]]
[1] 5.329071e-15
```

Advanced Programming in R: Functions (cont.)

Next we want to lapply function, lapply function is used to apply a function over a list so that is where you have 1 here. Lapply will always return a list which is of the same length as the input list. The syntax for the lapply is as follows. You have to use the command lapply and the list on which the function has to be applied and function which has to be applied on the list. Let us illustrate this lapply using example here create a matrix A which is having the elements from 1 to 9 which is of size 3 by 3 and create a matrix B which are having the elements 10 to 18 and which is of size 3 by 3.

Now, I am creating a list of matrices A and B using the list command. I want to evaluate the determinant of the matrices and then store them in a list. One way to do is calculator determinant of A, calculate determinant of B and calculated it and make them as a list. You can do easily the same operation using the lapply function which is shown here, I want to name that variable has determinant I use a function lapply, I want to apply the determinant function on my list when I do that it will calculate the determinant of A and then store it in the element 1 and calculate the determinant of B and store it in the element 2 of a list.

(Refer Slide Time: 08:19)

Data science for Engineers

mapply function

- **mapply** is a multivariate version of **lapply**.
- A function can be applied over several lists simultaneously.
- Syntax: `mapply(fun, list1, list2, ...)`

```
> source '~/volcylinder.R'
> dia=c(1,2,3,4)
> len=c(7,4,3,2)
> vol = mapply(volcylinder,dia,len)
> vol
[1] 5.497787 12.566371 21.205750 25.132741
```

Advanced Programming in R: Functions (cont.)

Now, let us move to the `mapply`. `mapply` is a multivariate version of `lapply`. What it does is this function can be applied on several lists simultaneously the syntax is `mapply` the function you need to apply on list 1 and list 2 together. So, we have seen R function `vol cylinder`. Now, let us suppose I want to calculate the volume for different diameters and different lengths which are having as a list here, I have a list of diameters and I have a list of lengths I want to evaluate the volume for each individual pairs of dia and length.

You can individually take this length and dia and execute the same function, but it is so, tedious `mapply` helps in simplify this job for us you need to create one variable `vol` and then apply the function `mapply` on the `vol cylinder` function because this is the function first we need to apply the function and then the list 1 and list 2. We have two lists `dia` and `length`, and what it does is it will take a pair and then calculate the volume and it will written the volumes were two lists of `dia` and `length` that are given.

(Refer Slide Time: 09:40)

The slide has a header 'Data science for Engineers' and a title 'tapply function'. In the R console, the following code is run:

```
> Id = c(1,1,1,1,2,2,2,3,3)
> Values = c(1,2,3,4,5,6,7,8,9)
> tapply(Values,Id,sum)
 1 2 3
10 18 17
```

The output is a table:

Id	val	sum
1	1	10
1	2	
1	3	
1	4	
2	5	18
2	6	
2	7	
3	8	17
3	9	

Below the slide, a man in a blue shirt is speaking.

Next we move on to the tapply function. tapply is used to apply a function over a subset of vectors given by combination of factors. The syntax for that is tapply a vector, practice and the function unit of length. So, to illustrate tapply let us use this example, I am creating a vector called Id using this concatenation operator which contains four 1's, three 2's and two 3's and I am also creating another vector which is having the values again a concatenation which are having the elements 1 to 9.

Now, if I want to add the values which are having the same ids tapply function can help me. So, what I need to do is tapply I want to add this values the adding is given here as a function which is sum according to the Id they belong to. What it does is it will take the elements corresponding to one Id for example, four 1's and the values 1 2 3 4 and sums them up $4 + 3, 7 + 2, 9 + 1$, so sum is 10. Similarly it will take the values corresponding to the Id 2 and then sum it up and values corresponding to the Id 3 and sum it up so that it will print the outputs which are indicating the sums of the elements of category 1, category 2 and category 3. The tapply prints the output as the sums of the elements which are having Id 1, Id 2 and Id 3.

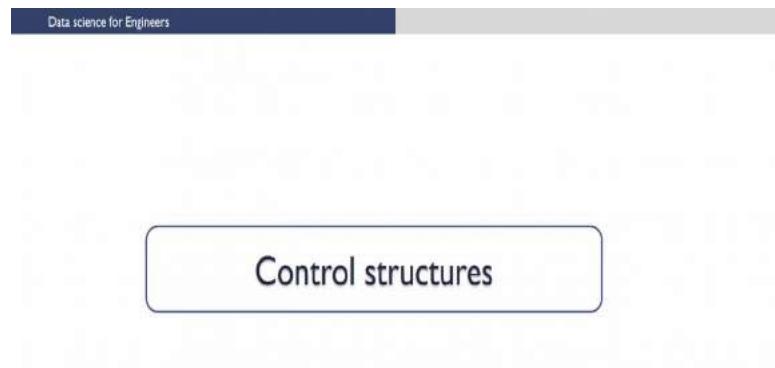
In this lecture we have seen how to write functions which takes multiple inputs and multiple outputs. We have seen inline functions and we have also seen some functions that are useful to loop over the objects. In the next lecture we are going to discuss about the control structures in R.

Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

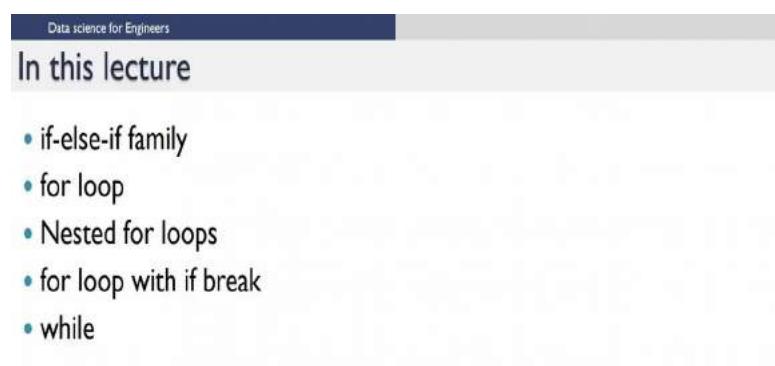
Lecture - 10
Control structures

(Refer Slide Time: 00:12)



Welcome to lecture 9 in the R module of the course, data science for engineers. Here we will look at the control structures in R.

(Refer Slide Time: 00:25)



- if-else-if family
- for loop
- Nested for loops
- for loop with if break
- while



In this lecture we are going to talk about if else if family constructs, for loop nested for loops, for loop with if break and while loop. Control structures can be divided into 2 categories.

(Refer Slide Time: 00:38)

Data science for Engineers

Control structures

- Execute certain commands only when certain condition(s) is satisfied (if-then-else)
- Execute certain commands repeatedly and use a certain logic to stop the iteration (for, while loops)

Programming in R – Control Structures

The first one is where you need to execute certain commands only when certain conditions are satisfied and example of this control structure is if then else type of constructs. The second one is execute certain commands repeatedly, and use certain logic to stop the iteration examples for this kind of constructs are for and while loops.

(Refer Slide Time: 01:04)

Data science for Engineers

If else family of constructs

If , If else and If-elseif - else are a family of constructs where:

- A condition is first checked, if it is satisfied then operations are performed
- If condition is not satisfied, code exits construct or moves on to other options

If construct

```
if(condition) {  
    statements  
}
```

If-else construct

```
if(condition) {  
    statements  
} else {  
    alternate statements  
}
```

If-else if-else construct

```
if(condition) {  
    statements  
} else if(condition) {  
    alternate statements  
} else {  
    alternate statements  
}
```

Programming in R – Control Structures

First look at if construct, what if does is if checks for a condition if the condition is satisfied it will execute the statements that are in the if loop. The next level is if else construct, where we want to do certain operations if the condition is satisfied and if not, we want to do some alternate operations.

So, the if else construct syntax looks like this, if the condition is satisfied perform this statements else perform this alternate statements. The next level is if else if else construct. So, here you will have 2 things if a condition is satisfied execute this statement; else if you check for another condition if that condition is satisfied execute this alternate statements, if both of them are not satisfied then do something else so the syntax is as follows; to illustrate this if let us consider this example here.

(Refer Slide Time: 02:09)

If else family of constructs: Example

IN THE EXAMPLE BELOW:

- IF x is greater than 7 operations inside the curved braces would occur
- Else the next condition i.e. x >8 would be checked, if this too is not true
- Final else condition is checked and if that too is false, there is no change in 'x'

<u>Code</u>	<u>Console Output</u>
<pre># If-elseif - else example x=6 if(x>7){ x=x+1 }else if(x>8){ x=x+2 }else { x=x+3}</pre>	<pre>> # If-elseif - else example > x=6 > if(x>7){ + x=x+1 + }else if(x>8){ + x=x+2 + }else { + x=x+3 + } > x [1] 9,</pre>

Programming in R - Control Structures

So, we have assigned a value of 6 to x, we are checking if x is greater than 7 because your value is 6 and we are checking a condition 6 is greater than 7 which is false this statement will not be executed. Now it will check whether the next condition, x is greater than 8; again this condition also fails because 6 is not greater than 8 and this part is not executed, since this 2 parts are not executed it will move to the else and then it will increment the value of x by 3; that means, you have now x as 6, $6 + 3 = 9$ and the value of 9 is assigned to x and this piece of code is executed in R you can see that the output is 9.

(Refer Slide Time: 03:00)

Data science for Engineers

Sequence function

- A sequence is one of the components of a 'for loop'
- Sequence function syntax :`seq(from,to,by,length)`
- Creates equi-spaced points between 'from' and 'to'

Parameter	Description	Console Output
from	starting number	
to	ending number	
by	increment or decrement (width)	
length	Number of elements required	<pre>Console ~ / ↻ > seq(from=1,to=10,by=3) [1] 1 4 7 10 > seq(from=1,to=10,length=4) [1] 1 4 7 10 > seq(from=1,to=10,by=4) [1] 1 5 9 > </pre>

Next, we move on to the for construct to understand the for function we need to understand what is a sequence function first. So, let us see what is a sequence function, sequence is one of the components of the for loop that is the reason why we are looking at the sequence function now, sequence function syntax is as follows.

Sequence function contains from the starting number from which the sequence has to begin to the ending number with which the sequence has to begin, you can define the sequence by either providing the by or length, when you provide this argument by it will specify by what increment or decrement the sequence has to be generated, when you provide this argument length. So, what it does is it will create number of elements that are required from the starting number to the ending number you can see the examples here, let us now assume that I want to create a sequence from 1 to 10 and then I want the width of 3.

So, the argument which I want to pass is by = 3, this will create one separated by 3 and then 4 4 and it leaves again 3 values and then 7 and then it leaves another 3 values and then A₁₀ instead I can do the same by specifying the length. So, the way you can do that is as follows. I want to generate a sequence from 1 to 10 which contains the 4 elements. So, it will generate the same thing so it will take from one and start from 1 and go up to 10 which contains 4 elements. Now if I want to say I want to create a sequence from 1 to 10 which is having a width of 4 this is how the output looks.

(Refer Slide Time: 04:50)

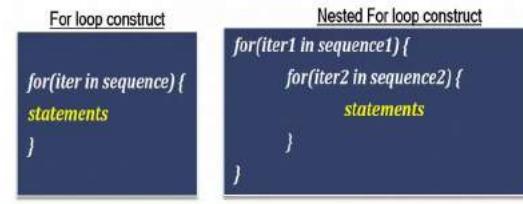
Data science for Engineers

for loop, Nested for Loops

The structure of a for loop construct comprises:

- A 'sequence' which could be a vector or a list
- 'iter' is an element of the sequence
- Statements

Nested for-loop : one or more for loop constructs are located within another.



Programming in R – Control Structures

Now let us move on to the for loop the structure of for loop construct comprises of a sequence which could be a vector or a list an iter which is an element of the sequence and the statements that are needed to be executed.

So, if you see the structure of this for loop, iter in a sequence as we seen iter is an element in the sequence the sequence is a list R vector; for every element in this sequence execute this statements is what this for loop construct is saying. So, next level of the for loops is a nested for loop. When you say nested for loop it means we have one or more for loop constructs that are located within another for loop.

The structure of the nested for loop is as follows, the for loop here is an inner for loop and the for loop, outside is called outer for loop for every iter 1 in the sequence 1 this for loop will get executed , it will go to the for loop 2 where it will perform this operations on sequence 2 for every iter 2 and return the output to illustrate this for loop.

(Refer Slide Time: 05:58)

The slide is titled "for loop: Example". It contains the following text:

Open a script file and type the following statements. Save the file as "forloop" and execute the script file

The value of 'sum' keeps changing inside the loop

The R code shown is:

```
1 n=5
2 sum=0
3 for(i in seq(1,n,1)){
4   sum = sum +i
5   print(c(i,sum))
6 }
```

A callout box highlights "n=5" and "sum=0" with the text "Initializing sum=0" and "loop variable". To the right is a table showing the state of variables at each iteration:

Loop variable (i)	sum
1	1
2	3
3	6
4	10
5	15

A video player interface is visible at the bottom, showing a progress bar and the title "Programming in R – Control Structures". A small video thumbnail of a person is also present.

We will have an example here, I am initialising number of elements to be 5 and sum to be 0. I am having a sequence which is starting from 1 and then ending at 5 with a width of 1, what I am doing inside the for loop is I am assigning sum + i value to the variable sum inside this for loop and I am printing the i which is the iter value in the loop and the sum when you execute this function this is how it behaves.

So, you have $n = 5$ it will keep it in memory. So, when we execute and $sum = 0$ it will initialize the sum to 0, for the first time it enters into the loop it will take value of 1 from the sequence. And then you have already sum as 0, $0 + 1$ is 1 and it will assign value 1 to the sum. You can see that in the first iter or a first iteration the value of sum is 1. In the second iteration you have the value sum as one it will go to the next iteration; that is now the i value is 2. So, you have already sum value as one and i is 2, $1 + 2$ is 3 and the value of 3 is assigned to the variable sum in the second iteration, you can see that value of the sum is 3 here and so on.

Since the sequence runs up to 5 the sum will be 15 at the end of 5 iterations, sometimes it is necessary to stop the function when you feel that the required condition is satisfied.

(Refer Slide Time: 07:41)

The slide is titled "For- loop with if-break". It contains the following text in a callout box:

A "break" statement once executed, program exits the loop even before iterations are complete
"break" command comes out of the innermost loop for nested loops

The R code shown is:

```
1 n=100
2 sum=0
3 for(i in seq(1,n,1)){
4   sum = sum + i
5   print(c(i,sum))
6   if(sum>15){
7     break
8 }}
```

A yellow arrow points from the "break" keyword in the code to a note below it.

The note says: "break" command, the loop is terminated after the 6th iteration

To the right of the code is a table showing the state of variables for each iteration:

Loop variable (i)	sum	Condition (sum>SUM)
1	1	False
2	3	False
3	6	False
4	10	False
5	15	False
6	21	True

A small video player interface is visible at the bottom right.

This can be achieved using a break statement in the for loop. So, let us see how to do this. I am assigning a variable value of 100 to the variable n and I am initializing the sum as 0. Now I want to stop the loop when the sum exceeds 15, how do I do that? So, in the for loop what you have to have is if break construct. So, in the for loop these are the statements that are available for every iteration I am adding this sum and then iter value and assigning it to the sum, and I am printing the vector which is containing the loop variable and sum.

And I have to check a condition if the sum is greater than 15 I will say break because this is the condition which I want to break the loop. So, this break statement once executed the program exit the loop even before the iterations are complete. So now, let us see how this things work so in the first iteration the loop variable has a value 1 because it is a sequence starting from 1 and the last value is 100 here. So, we have seen in the previous example also for the first time the value of the sum becomes 1.

And it checks the condition if sum is greater than 15 because sum value is 1 it is not greater than 15 the break statement will not be executed. And the break statement will not be executed until 5 iterations when the iteration number 6 comes sum value is already 15, and the iteration value now is 6, 15 + 6 will become 21 and that 21 will get assigned to the variable sum.

Now if this condition is checked if sum is greater than 15; this condition is satisfied and the break statement get executed; once this break statement get executed the program quit from the for loop next we move on to another construct which is while loop.

(Refer Slide Time: 09:43)

The slide has a header 'Data science for Engineers' and a title 'While loop'. It contains the following text and code:

A while loop is used whenever you want to execute statements until a specific condition is violated

Consider the sequence of natural numbers.

What is the value of the natural number up to which the calculated sum is less than specified "Fin_sum"?

$$1+2+3+\dots+n = \text{Fin_sum} (15)$$

```

1 sum=0 } Initialize the
2 i=0 variables
3 Fin_sum=15
4 while(sum<Fin_sum){
5   i=i+1
6   sum=sum+i
7   print(c(i,sum))
8 }
```

Loop variable (i)	sum	Condition (sum<Fin_sum)
1	1	True
2	3	True
3	6	True
4	10	True
5	15	False

Programming in R – Control Structures

A while loop is used whenever you want to execute statements until a specific condition is violated, we can see it as an akin to for loop with if break construct.

Let us consider a sequence of natural numbers; you want to find a natural number n after which the sum of the natural numbers n is greater than certain final sum you wanted to be. You will consider the same example as we have seen I am initializing the sum as 0 and the initial variable $i = 0$. And I want the final sum to be 15. This is how I write a while function while sum is less than final sum. I want to increment the i value by 1 and then reassign it to the value i and I also want to increase the sum by the iter value and then reassign it to sum and then finally, print the iteration value and the sum. These commands get executed until the loop variable has a value 4.

Let us understand how does it works; for the first time i is 0 it will check the condition sum is less than final sum, the condition is true what it does is it will increment the i by 1. So, $i + 1$ which is $0 + 1$ you will get the loop variable as 1 and the sum is $0 + 1$. So, sum as the so gets the a value 1. Now this statement prints this first line, now it will go to the next iteration. Now it checks whether the sum is less than final sum; the sum is 1 which is less than the final sum 15 it will go for the next iteration, it will update the value

of loop variable and the value of sum variable. At the fifth iteration what it does is you have a sum variable 15 which = 15, but not less than 15. This statement is false and it will come out of the loop. So, in this lecture we have seen how the if else family of constructs can be coded in R and how to code for loops and while loops in R.

In the next lecture we are going to see how to perform basic graphics operations in R.

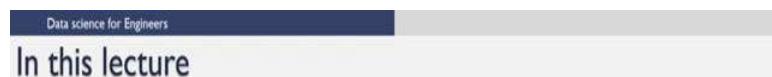
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 11
Data visualization in R Basic graphics

Welcome to the lecture 10 and the final lecture in the R module of the course data science for engineers. In the previous lectures, we have seen; what are the basic data types that are supported by R, how to write scripts, how to write functions and how to do control structures, how to do programming and so on.

(Refer Slide Time: 00:35)

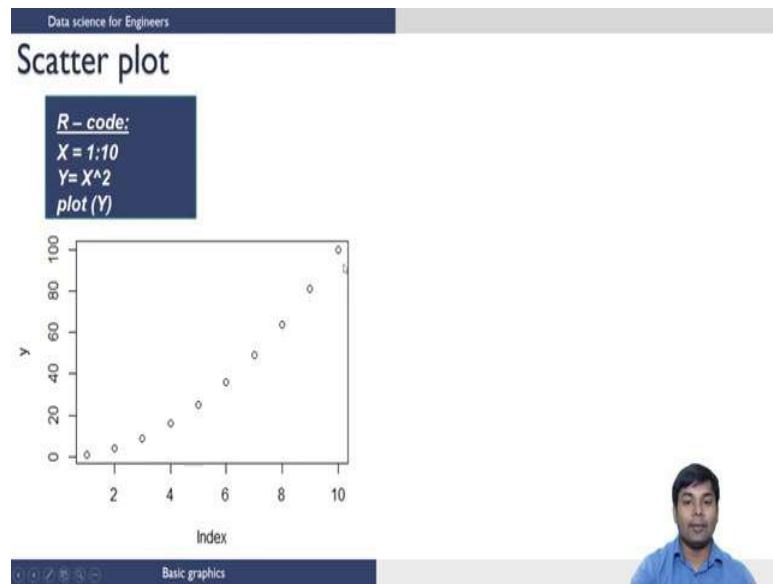
Data science for Engineers
In this lecture

- Basic graphics
 - Scatter
 - Line
 - Bar
- Need for sophisticated graphics



In this lecture, we are going to show you how to generate some basic graphics; such as scatter plot, line plot and bar plot using R, and we will also give a brief idea on why there is a need for more sophisticated graphics and how R does it.

(Refer Slide Time: 00:54)



First we will consider scatter plot; scatter plot is one of the most widely used plots where we have some independent variable and dependent variable, when you want to see how a dependent variable is dependent on the independent variable. We can use scatter plot generating the scatter plot in R, is quite simple. The first command here shows, it is creating a vector which is having the elements from 1 to 10, and the next command here takes this x and calculates the element wise square of the x and then assign it to value y.

When you plot y it will generate this plot here. Since we have not specified, what is x which is independent variable, the R generates its own independent variable as the index, since this vector contains 10 elements. It will create the index based on the number of elements in the vector and then the y values which are the squares of elements 1 to 10 that are 1 4 9 and so on are shown in the y axis, and $10^2 = 100$. We have the final value here on the y axis as 100.

(Refer Slide Time: 02:24)

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models).

Usage
mtcars

Format
A data frame with 32 observations on 11 variables.

[. 1] mpg Miles/(US) gallon
[. 2] cyl Number of cylinders
[. 3] disp Displacement (cu.in.)
[. 4] hp Gross horsepower
[. 5] drat Rear axle ratio
[. 6] wt Weight (1000 lbs)
[. 7] qsec 1/4 mile time
[. 8] vs V/S
[. 9] am Transmission (0 = automatic, 1 = manual)
[.10] gear Number of forward gears
[.11] carb Number of carburetors

Source
Henderson and Velleman (1981). Building multiple regression models interactively. *Biometrics*, 37, 391-411.

Basic graphics

So, let us illustrate the scatter plot using some inbuilt data set; that is available in R. So, we are talking about a data set by name empty cars. So, you can access this data set by just typing empty cars. This data set is a data frame which contains 32 observations on 11 variables. The variables are listed here such as number of cylinders, which is represented by variable c y l and m p g. What is the mileage that this cars gives; that is miles per us gallon and weight w t, which is weight of the car and so on.

(Refer Slide Time: 03:13)

R - code :

```
plot(mtcars$wt, mtcars$mpg, main="Scatterplot Example", xlab="Car Weight", ylab="Miles Per Gallon", pch=19)
```

Corresponds to different shapes for points, for more such options check 'graphics parameters' in help

Scatterplot Example

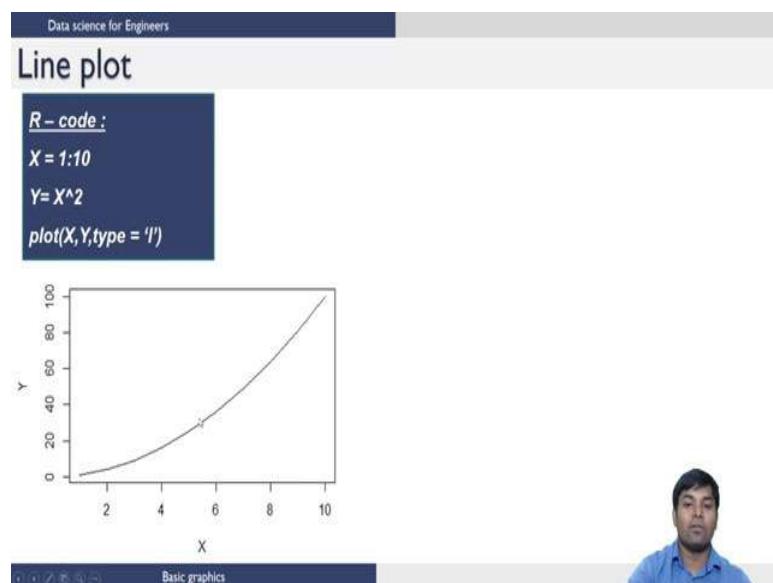
Miles Per Gallon

Car Weight

Basic graphics

Now, let us try to plot a scatter plot between weight and mpg of this data frame . To do that what we need to use, is plot command. This is your independent variable car weight, this is your dependent variable miles per gallon and this main helps in naming the title of the graph x lab to give a label for x axis, y lab is used to give a label for y axis and the this pch corresponds to different shapes for points, and this pch = 19 corresponds to the shape that is shown in this screen. You can use different pch values to obtain different shapes for the points in a scatter plot.

(Refer Slide Time: 04:02)



Next we move the line plot. You can take same example what we have seen earlier. If the same plot command can be used what you need to do to generate a line plot, is to specify an extra argument type which is l.

(Refer Slide Time: 04:24)

Bar plot

Syntax:

```
barplot(H, names.arg, xlab, ylab, main, names.arg, col)
```

R – code :

```
H <- c(7,12,28,3,41)
M <- c("Mar", "Apr", "May", "Jun", "Jul")
barplot(H, names.arg = M, xlab = "Month", ylab = "Revenue",
       col = "blue", main = "Revenue chart", border = "red")
```



So, type = 1 generates a line, instead of the scatter plot. Next we move on to the bar plot, the syntax to generate bar plot in R is as follows; bar plot of h. These are the heights which can be a vector or matrices to keep it simple. We will deal with only vectors and names dot argument. What this argument does is? it will print the names under the each attribute in the H x lab and y lab, and main has a same meanings as what we have seen for the scatterplot, and colour gives us an option to give colour to the bar plot. This is the R code that can be used to generate the bar plot. I want to define h heights of the barcodes as a vector, which is having the values 7, 12, 28, 3 and 41, and I want to create another vector which is of character variable, which is having the values March, April, May, June and July, and now, I am trying to create a bar plot with h as heights and name start arguments as m x lab as month, y lab as revenue and the colour of the bar notes I want, is blue and the title is revenue chart and the border is red.

(Refer Slide Time: 05:48)



So, when you execute these commands, this is how the bar plot looks. These are the heights of the bar charts, this is A 3. And then in the names dot variable we have March, April, May, June and July, which is printed at the bottom of each height, and the x axis is month, y axis is revenue and the title is revenue chart.

(Refer Slide Time: 06:10)



Now, let us see why there is a need for sophisticated graphics. Let us say there is a need for you to show multiple plots in a single figure as shown below. How do you do this?

(Refer Slide Time: 06:21)

Data science for Engineers

Challenges

The exact figure as per the previous slide can be reproduced with the following code:

```
par(mfrow=c(2,4))
days <- c("Thur", "Fri", "Sat", "Sun")
sexes <- unique(tips$sex)
for (i in 1:length(sexes)) {
  for (j in 1:length(days)) {
    currdata <- tips[tips$day == days[j] & tips$sex == sexes[i],]
    plot(currdata$total_bill, currdata$tip/currdata$total_bill,
         main=paste(days[j], sexes[i], sep=", "), ylim=c(0,0.7), las=1)
  }
}
```

Basic graphics



What are the challenges that you face when you want to create figure that was shown in the earlier slide. So, the exact figure can be reproduced using this code which is shown here for this.

(Refer Slide Time: 06:37)

Data science for Engineers

Challenges

But the code requires work such as :

- Knowing when to introduce a for loop
- Which columns of the data.frame to select
- The positioning of each graph in the grid etc
- Less pleasing visuals

Basic graphics



What you have to know, is you have to know where to introduce for loop, which columns of data frame to be selected for plotting, and you have to also position each graph in the grid etcetera. Even though you do all of this operations, the visuals are less pleasing that is where we need more sophisticated graphics packages in R . This is where the ggplot2 comes into picture. The ggplot2 provides a very beautiful package for generating graphics in R in this course, we have not deal much with ggplot2.

(Refer Slide Time: 07:14)

Data science for Engineers

Summary

- 1) Scatter plots
- 2) Line plots
- 3) Bar plots
- 4) Challenges and disadvantages of basic graphics

Basic graphics

In summary, we have seen how to generate scatter plots, line plots and bar plots in the R. We have also seen the challenges and disadvantages of basic graphics and the need for using the advanced packages; such as ggplot2 for generating beautiful graphics in R. With this we end the R module for this course. Wish you all the best for the next modules in this course.

Thank you.

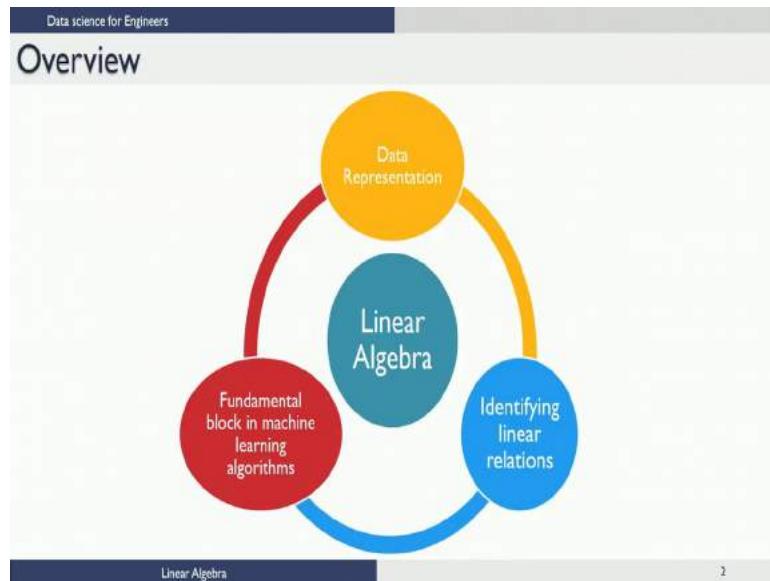
Data Science for Engineers
Prof. Raghunathan Rangaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 12
Linear Algebra for Data science

This lecture is on linear algebra for data science. Linear algebra is a very fundamental part of data science and usually a typical one semester linear algebra course will run for about 36 hours. What we are trying to do here is we are trying to introduce the use of linear algebra in data science in a few hours. So, that necessarily means that we cannot cover the topic of linear algebra in all its detail; however, what we have attempted to do here is to identify the most important concepts from linear algebra, that are useful in the field of data science and in particular for the material that we are going to teach in this course. So, in that sense we have crystallized a few important concepts from linear algebra that the participants should learn and understand.

So, that is one thing that I would like to mention right at the beginning. The second thing that I would like to mention is the following; Linear algebra can be treated very theoretically very formally; however, in the short module on linear algebra which has relevance to data science ,what we have done is we have tried to explain the ideas in as simple fashion as possible without being too formal. However, we do not do any hand waving we teach linear algebra in a simple fashion. So, that is another thing that I would like you to remember as we go through this material. So, we first start by explaining what linear algebra is useful for.

(Refer Slide Time: 02:06)



So, when one talks about data science, Data Representation becomes an important aspect of data science and data is represented usually in a matrix form and we are going to talk about this representation and concepts in matrices. The second important thing that one is interested from a data science perspective is, if this data contains several variables of interest, I would like to know how many of these variables are really important and if there are relationships between these variables and if there are these relationships, how does one un-cover these relationships?

So, that is also another interesting and important question that we need to answer from the viewpoint of understanding data. Linear algebraic tools allow us to understand this and that is something that we will teach in this course. The third block that we have basically says that the ideas from linear algebra become very very important in all kinds of machine learning algorithms.

So, one needs to have a good understanding of some of these concepts before you can go and understand more complicated or more complex machine learning algorithms. So, in that sense also linear algebra is an important component of data science. So, we will start with matrices. Many of you would have seen matrices before. I am going to look at matrices and summarize the most important ideas that are relevant from a data science viewpoint. What is a matrix? Matrix is a form of organizing data into rows and columns. Now, there are many ways in which you can organize data. A matrix provides you a convenient way of organizing this data.

(Refer Slide Time: 01:48)

Data science for Engineers

Matrix theory and linear algebra

Matrix Theory and Linear Algebra

- Matrices can be used to represent samples with multiple attributes in a compact form
- Matrices can also be used to represent linear equations in a compact and simple fashion
- Linear algebra provides tools to understand and manipulate matrices to derive useful knowledge from data

Linear Algebra

So, if you are an engineer and you are looking at data for multiple variables, at multiple times, how do you put this data together in a format that can be used later, is what a matrix is helpful for.

Now, matrices can be used to represent the data or in some cases matrices can also be used to represent equations and the matrix could have the coefficients in several equations as its component. Now, once we generate these matrices then you could use the linear algebra tools to understand and manipulate these matrices, so that you are able to derive useful information and useful knowledge from this data.

(Refer Slide Time: 04:44)

LINEAR ALGEBRA AND MATRICES

So, let us start and then try and understand how we can understand and study matrices.

(Refer Slide Time: 04:50)

- Usually matrices are used to store and represent the data on machines
- Matrix is a very natural approach for organizing data
- In general, data is organized in the following fashion
 - Rows represent samples
 - Columns represent the values of the variables (or attributes)
 - It is also possible to use rows for variables and columns for samples
 - However, we will stick to rows as samples and columns as variables in all of the material that will be presented

As I mentioned before matrices are usually used to store and represent data on machines and matrix is a very natural approach for organizing data. Typically when we have a matrix it is a rectangular structure with rows and columns. In general, we use the rows to represent samples and I will explain what I mean by this in subsequent slides and we use columns to represent the variables or attributes in the data.

Now, this is just one representation. It is possible that you might want to use rows to represent variables and columns to represent samples and there is nothing wrong with that; however, in this course and all the material that we present in this course we will stick to using rows to represent samples and columns to represent variables as far as this course is concerned.

(Refer Slide Time: 05:47)

Data science for Engineers

Data representation: Examples

- A real life example
 - Consider a reactor which needs to be controlled using multiple attributes from various sensors like Pressure (Pa), Temperature (K), Density (gm/m^3) etc.
 - Independently, the sensors have generated 1,000 data points
 - This complete set of information is contained in

Linear Algebra

$$\begin{matrix} & P & T & \rho \\ 1 & [300 & 300 & 1000] \\ \vdots & \vdots & \vdots & \vdots \\ 1000 & 500 & 1000 & 5000 \end{matrix}$$

Let me explain matrix using a real life example. Let us consider that you are an engineer and you are looking at a reactor which has multiple attributes and you are getting information from sensors such as pressure sensors, temperature sensors and density and so on.

Now let us assume that you have taken 1000 samples of these variables. Now you want to organize this data somehow. So that you can use it for purposes needed. One way to do this is to organize this in this matrix form, where the first column is the column that corresponds to the values of pressure at different sample points. The second column corresponds to the value of temperature at several sample points and the third column corresponds to the value of density at several sample points.

So, that is what I meant when I said the columns are used to represent the variable. So, each column represents a variable column 1 pressure, column 2 temperature and column 3 density and when you look at the rows; the first row represents the first sample.

Here in the first sample you will read that the value of pressure was 300, the value of temperature was 300 and the value of density was 1000. Similar to that you will have many rows corresponding to each sample point up to the last row 1000th row, which is a 1000 sample point; which has a pressure is 500 temperature is 1000 and density is 5000.

(Refer Slide Time: 07:25)

The slide is titled "Data representation: Examples". It contains the following content:

- Example 2:
 $X = [1,2,3]^T$
 $Y = [2,4,6]^T$
- X and Y are vectors pertaining to some attributes
- We define the A matrix using a column bind of X and Y thus representing data in a matrix format (the code for the same is attached)

R Code:

```
x=c(1,2,3)
y=c(2,4,6)
A=cbind(x,y)
print(A)
```

Output:

```
[1] 1 2
[2] 2 4
[3] 3 6
```

A video of a man speaking is shown on the right.

Let us take another example let us say I have 2 vectors, $X = [1, 2, 3]^T$ and $Y = [2, 4, 6]$. Let us say this is some variable that you have measured and Y is some other variable you have measured and the 3 values could represent the 3 sampling points at which you measured these.

Now, in R if you want to put these numbers into a matrix, it is a very simple code. What you do is: $X = c(1, 2, 3)$ it tells you it is a column vector with values 1, 2, 3 y is a column vector with values 2, 4, 6 and then you use the command $A = cbind(x, y)$ which puts these together and when you print A you get the value of this matrix.

(Refer Slide Time: 08:13)

The slide is titled "Data representation: Examples". It contains the following content:

- The simplicity in representation will become apparent when the image below is considered

A screenshot of a software interface is shown, with two red circles highlighting specific regions. The left circle highlights a grayscale image of a textured surface. The right circle highlights a 3x3 grid of numerical values in a table.

A video of a man speaking is shown on the right.

Now, we have been talking about using matrices to represent data from engineering processes sensors and so on. The notion of matrix and manipulating matrices is very important for all kinds of applications. Here is another example where I am showing how a computer might store data about pictures. So, for example, if you take this picture here on this left hand side and you want to represent this picture somehow in a computer. One way to do that would be to represent this picture as a matrix of numbers.

So, in this case for example, if you take a small region here you can break this small region into multiple pixels and depending on whether a pixel is a white background or a black background you can put in a number. So, for example, here you see these numbers which are large numbers which represent white background and you have these small numbers which represent black background. So, this would be a snapshot or a very small part of this picture.

Now, when you make this picture into many such parts you will have a much larger matrix and that larger matrix will start representing the picture. Now, you might ask; why would I do something like that? There are many many applications where you want the computer to be able to look at different pictures and then see whether they are different or the same or identify sub components in the picture and so on. And all of those are done through some form of matrix manipulation and this is how you convert the picture into a matrix.

Now notice that while we converted this matrix we have again got into a rectangular form, where we have rows and columns, where data is filled as a representation for this picture.

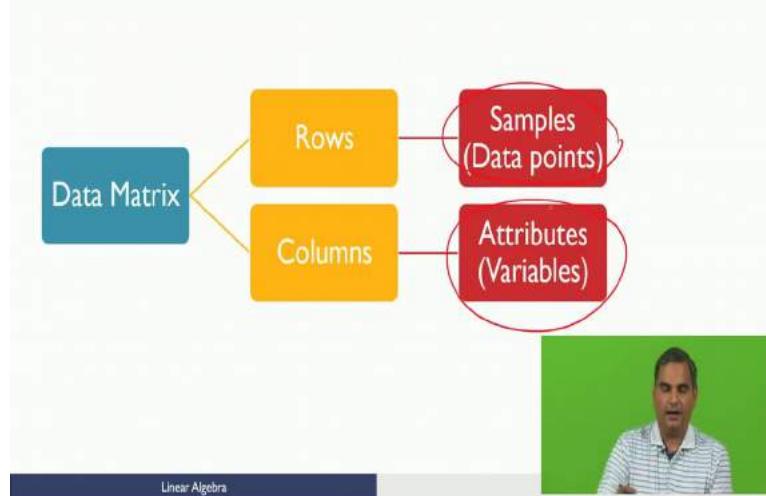
(Refer Slide Time: 10:07)

- Storing
 - The image is stored in the machine as a large matrix of pixel values across the image.
 - Thus, storing the pixel value matrix is equivalent to storing the image for the machine
- Identification
 - Several machine learning algorithms are deployed in order to “teach” the machine how to identify a particular image.
 - Linear algebra and matrix operations are at the heart of these machine learning algorithms.

So, the image that I showed before could be stored in the machine as a large matrix of pixel values across the image. And you could show other pictures and

then say are these pictures similar to this, are these dissimilar, how similar or dissimilar and so on and the ideas from linear algebra matrix operations are at the heart of these machine learning algorithms.

(Refer Slide Time: 10:32)



So, in summary if you have a data matrix- the data matrix could be data from census in an engineering plan. It could be data which represents a picture; it could be data which is representing the model where you have the coefficients from several equations. So, the matrix basically could have data from various different sources or various different viewpoints and each data matrix is characterized by rows and columns and the rows represent samples and the columns represents attributes or variables.

(Refer Slide Time: 11:14)



• **IDENTIFICATION OF
INDEPENDENT ATTRIBUTES**



Now that we have understood how we generate a matrix and why we generate matrices. The next question might be the following, supposing I have a matrix where I have several samples and several variables, I might be interested in knowing if all the variables that are there in the data are important. In other words, I would really like to know of all these variables, how many are actually independent variables?

So that I know how much information is there in this data. So, if let us say I have thousands of variables and of those there are only 4 or 5 that are independent. Then it means that actually I can store values for only these few variables and calculate the remaining as a function of these variables. So, it is important to know how much information I actually have.

(Refer Slide Time: 12:08)

Data science for Engineers

Further analysis

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

Linear Algebra

So, this would lead to the following questions. The first question might be; Are all the attributes or variables in the data matrix really relevant or important? Now a sub question is to say are they related to each other. If I can write one variable as a combination of other variables then basically I can drop that variable and retain the other variables and calculate this variable whenever I want.

So, that is a very important idea that we want to use in machine learning and various other applications. So, how do I find out how many of these variables are really independent and let us assume that I do and that only a few variables are really independent, then how do I identify the relationship between these variables and the other dependent variables and once I do that how do we actually reduce the size of the data matrix and so on; are questions that one might be interested in answering.

(Refer Slide Time: 13:09)

Data science for Engineers

Identification of independent attributes: Example

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

Linear Algebra

13

So, let us consider the example that we talked about; the reactor with multiple attributes. In the previous slide, we talked about pressure, temperature and density. Here I have also included viscosity. Let us say I have 500 samples. Then when I organize this data with the variables in the columns and samples in the row, then I will get a 500 by 4 matrix, where each row represents one of the 500 samples and if you go across the column, it will represent the variable values, at all the samples that we have taken. Now, I want to know how many of these are really independent attributes.

(Refer Slide Time: 13:51)

Data science for Engineers

Identification of independent attributes: Example

- Domain knowledge
$$D \sim f(P, T)$$
- Thus, in some sense D is a function of P and T
- Implying that at least one attribute is dependent on the others
- This variable can be calculated as a linear combination of the other variables
- The physics of the problem helps us identify the relationship in the data matrix
- We now ask if the data itself will help us identify these relationships

Linear Algebra

14

So, from domain knowledge it might be possible to say that density is in general a function of pressure and temperature. So, this implies that at least one attribute is dependent on the other and if this relationship happens to be a linear relationship then this variable can be calculated as a linear combination of the other variables.

Now, if all of this is true then the physics of the problem has helped us identify the relationship in this data matrix. The real question that we are interested in asking is if the data itself can help us identify these relationships.

(Refer Slide Time: 14:26)

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

Let us first assume that we have many more samples than attributes for now and once we have the matrix, when we want to identify the number of independent variables. The concept that is useful is the rank of the matrix and the rank of the matrix is defined as the number of linearly independent rows or columns that exist in the matrix.

And once you identify these number of linearly independent rows or columns then you could basically say that I have only so many independent variables and the remaining are dependent variables and the rank of the matrix can be easily found using the rank command in our rank of A.

(Refer Slide Time: 15:08)

- Consider another example

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

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

R Code

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

Output

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

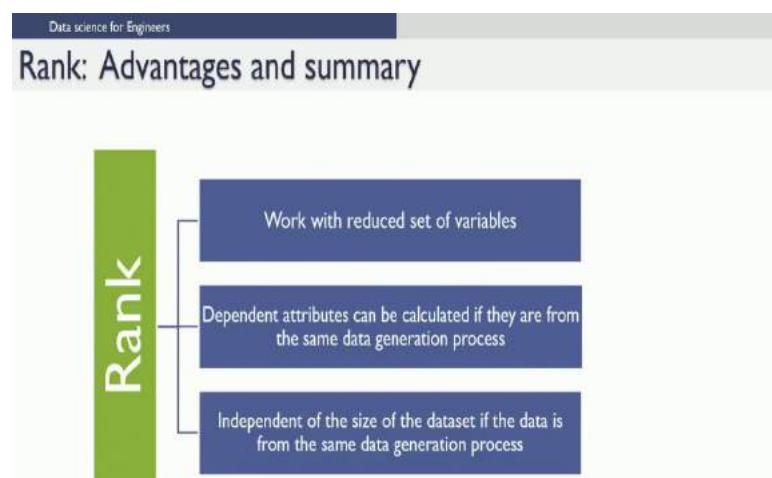


Linear Algebra
17

So, consider this example here where I have this a matrix which is 1, 2, 3, 2, 4, 6, 1, 0, 0. If you notice this matrix has been deliberately generated such that the second column is twice column one.

So, in other words the second column is dependent on the first column or you could say the first column is dependent on the second column. Now, there is one other column which is independent of these two so, if you think about this matrix there are 2 independent columns; which basically means there are 2 independent variables. So, if you were to use R to identify this, simply load the correct library and then use the command rank of A and you will get the rank of the matrix to be 2.

(Refer Slide Time: 16:00)

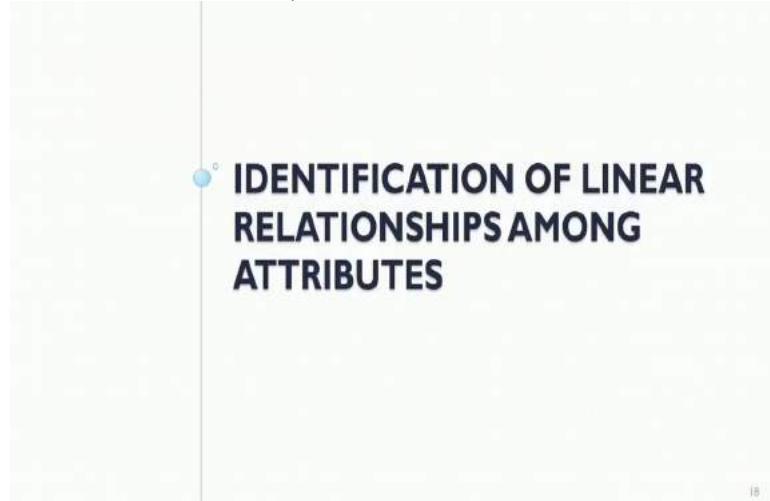


So, the notion of rank is important. It allows us to work with a reduced set of variables and once we identify the independent

variables, the dependent variables or attributes can be calculated from the independent variables, if the data is being generated from the same data generation process.

And if you identify that there are certain variables which are dependent on other variables and as long as the data generation process is the same, it does not matter how many samples that you generate, you can always find the de-pendent variables as a function of the independent variables.

(Refer Slide Time: 16:45)



Now that we have talked about identifying how many independent variables are there; assume that the number of independent variables are less than the number of variables. Then that basically automatically means that there are really linear relationships between these variables.

(Refer Slide Time: 17:13)

- Now that we have identified the number of linearly independent attributes:
 - How does one identify those linear relations among the attributes?
- Such questions are addressed by the linear algebraic concepts of null space and nullity



Now we ask the next question as to how do we identify these linear relationships among variables or attributes. So, this question of how does one identify the linear relationships among attributes, is answered by the concepts of null space and nullity which is what we going to describe now.

(Refer Slide Time: 17:25)

Data science for Engineers

Null space for data science

- The null space of a matrix A consists of all vectors β such that $A\beta = \mathbf{0}$ and $\beta \neq \mathbf{0}$
- Nullity of a matrix is the number of vectors in the null space of the given matrix
- The size of the null space of a matrix provides us with the number of linear relations among the attributes
- And the null space vectors β are useful to identify these linear relationships

$$A_{3 \times 3} \beta_{3 \times 1} = \mathbf{0}_{3 \times 1} \quad \text{②}$$

When we have a matrix A and if we are able to find vectors β such that $A\beta = 0$ and $\beta \neq 0$ then we would call this vector β as being the null space of the matrix. So, let us do some simple numbers here for example, if A is a 3 by 3 matrix because β multiplies A , β has to be 3 by 1 and the resultant will be some 3 by 1 vector and if all the elements of this 3 by 1 vector are 0, then we would call this β as being the null space of the matrix. Now interestingly the size of the null space of the matrix provides us with the number of relationships that are among the variables.

If you have a matrix which is of dimension 5 and let us say the size of null space is 2, then this basically means that there are 2 relationships among these 5 variables, which also automatically means that of these 5 variables only 3 are linearly independent because the 2 relationships would let you calculate the dependent variables as a function of these independent variables.

(Refer Slide Time: 18:46)

Data science for Engineers

Null space : general description

- Let us suppose
- $A = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$ is a data matrix and there is one vector in the null space of A , i.e., $\beta = [\beta_1 \dots \beta_m]^T$, then as per the definition, β satisfies all the equations given below

$x_{11}\beta_1 + x_{12}\beta_2 + \dots + x_{1n}\beta_n = 0$

- $x_{11}\beta_1 + x_{12}\beta_2 + \dots + x_{1n}\beta_n = 0$
- \vdots
- $x_{m1}\beta_1 + x_{m2}\beta_2 + \dots + x_{mn}\beta_n = 0$

Now, let us look at this in little more detail to understand how we can use this null space vectors. Let us assume that I have a matrix such as this. Now if there is a β which is what we have written here such that $A\beta = 0$. As I mentioned in the previous slide, if I have this matrix of this dimension and if I multiply β with this matrix, then on the right hand side there are going to be several elements and for β to be the null space of this matrix every one of the elements has to be equal to 0. Now, let us look at what each of these elements are equal to.

So, if you take the first element on the right hand side, that would be a product of the first row of this data matrix and this β vector which

would basically be $x_{11}\beta_1 + x_{12}\beta_2$; all the way up to x_n and $\beta_n = 0$. Now, similarly if you get to the second row and multiply the second row by this vector you will get another equation.

So, if we keep going down, for every sample if you write this product, you are going to get an equation. The last sample for example, will be $x_{m1}\beta_1 + x_{m2}\beta_2 + \dots + x_{mn}\beta_n = 0$. Now, there is something interesting that you should notice here. This equation seems to be satisfied for all samples. So, what this basically means is, irrespective of the sample, the variables seem to hold this equation and since this equation is held for every sample we would assume that this is a true relationship between all of these variables or attribute. So, in other words this β_1 to β_m gives you in some sense a model equation or a relationship among these variables.

So, one might say that this equation can generally be written as $x_1\beta_1 + x_2\beta_2$ all the way up to $x_n\beta_n = 0$; where you can take any sample and substitute the values of the variables in that sample at $x_1 x_2$ up to x_n and this is to be satisfied. So, this is a true relationship.

(Refer Slide Time: 21:19)

- Notice that if $A\beta = 0$, every row of A when multiplied by β goes to zero
- This implies that variable values in each sample (represented by a row) behave the same
- This helps in identifying the linear relationships in the attributes
- Every null space vector corresponds to one linear relationship
- This idea is demonstrated further using examples

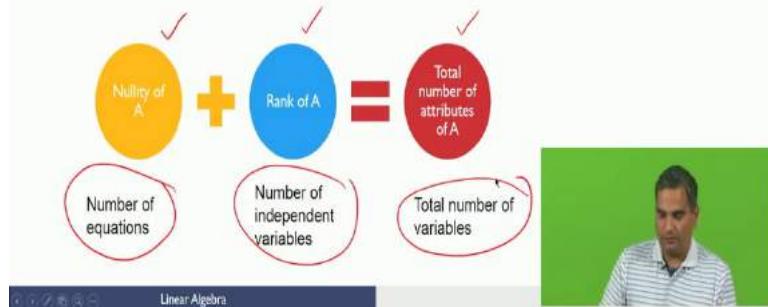
So, this is what we have said. Again here notice that if $A\beta = 0$ every row of A when multiplied by β goes to 0.

So, this implies that the variable values in each sample behave the same so, we have truly identified a linear relationship between these variables. Now, every null space vector corresponds to one such relationship and if you have more vectors in the null space then you have more relationships that you can uncover.

(Refer Slide Time: 21:49)

Rank nullity theorem

- Consider the data matrix A with the null space and nullity as defined before
- The rank-nullity theorem helps us to relate the nullity of the data matrix to the rank and the number of attributes in the data
- According to the rank-nullity theorem

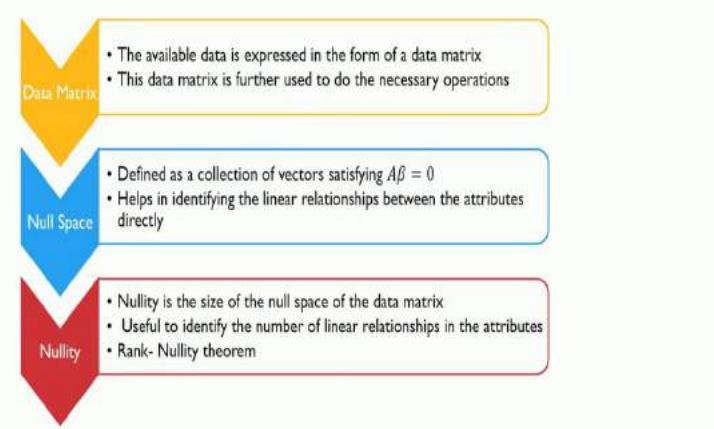


Linear Algebra

So, we will demonstrate this example this with a further example. So, this rank nullity theorem basically says the nullity of matrix A + the rank of matrix A is going to be equal to the total number of attributes of A or the number of columns of the matrix. So, the nullity of a tells you how many equations are there; there are so, many vectors in the null space.

The rank of A tells you how many independent variables are there and when you add these two you should get the total number of variables that is there in your problem.

(Refer Slide Time: 22:29)



Linear Algebra

24

So, to summarize, when you have data, the available data can be expressed in the form of a data matrix and as we saw in this lecture this

data matrix can be further used to do different types of operations. We also defined null space, null space is defined as a collection of vectors that satisfy this relationship A times $\beta = 0$.

So, this basically helps in identifying the linear relationships between the attributes or the variables directly and the number of such vectors or number of such relationships is what is given by the nullity. The Nullity of the matrix tells you how many relationships are there or how many vectors are there in the null space.

(Refer Slide Time: 23:13)

Data science for Engineers

Null space: An Example

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

1	2
3	4
5	6

Number of columns in A = 2

Rank of A = 2

Thus, nullity = 0

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

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

Console output

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

Console output

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

Linear Algebra 25

Let us take some examples to make these ideas little more concrete. Let us take a matrix A which is 1, 3, 5, 2, 4, 6. A quick look at this matrix and the numbers would tell you that these two columns are linearly independent and subsequently because these columns are linearly independent there can be no relationships among these two variables.

So, you can see that the number of columns 2 since they are independent the rank is 2. Since the rank is 2, nullity is 0 and because both the variables are independent you cannot find a relationship. If you were able to find a relationship then the rank should not have been 2. So, this basically implies that null space of the matrix A does not contain any vectors and as we mentioned before these variables are linearly independent.

Now, if you want to do the same thing in R what you do is you define the matrix A which basically is done using this command. This n columns equal to 2 tells you how many columns this number should be put in. So, since there are two columns, these numbers will be partitioned into 1, 3, 5, 2, 4, 6 and as we saw before you can actually get the rank of A.

And you can print the number of columns, you can print the rank and you can print nullity which is the difference between columns and the rank number of columns and the rank.

(Refer Slide Time: 24:43)

- Now consider A with attributes

$\{x_1, x_2, x_3\}$ such that

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

Number of columns in A = 3

Rank of A = 2

Thus, nullity = 1

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

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

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



Linear Algebra

Now, let us take the other example that we talked about where I mentioned that we have deliberately made the second column twice the first column. So, in this case as we saw before the rank of the matrix would be 2 because there are only two linearly independent columns and since the number of variables = 3, nullity will be $3 - 2 = 1$.

So, when we look at the null space vector you will have one vector which will identify the relationship between these three variables.

(Refer Slide Time: 25:15)

$$A\beta = 0$$

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

$A\beta = 0$

$b_1 + 2b_2 = 0$

$b_3 = 0$

- Thus we obtain,

$$b_1 + 2b_2 = 0$$

$$b_3 = 0$$

$b_1 = -2b_2$

- The null vector is $B = [b_1 \ b_2 \ b_3]^T = [-2b_2 \ b_2 \ 0]^T = k[-2 \ 1 \ 0]^T$

- We see that we obtain a direct linear relationship between the attributes of A using null space and rank-nullity theorem

- The same concept can be extended for bigger data set

$-2x_1 + x_2 = 0$

Linear Algebra

17

So, to understand how to calculate the null space let us look at this example. So, we set up this equation $A \beta = 0$ and we know we will get only 1 β here and β we have written as $b_1 b_2 b_3$ and when we do the first row versus column multiplication I will get $b_1 + 2b_2 = 0$.

When I do the second row and column multiplication I will get $2b_1 + 4b_2 = 0$ and when you do the third multiplication you get $b_3 = 0$. Now the second equation which is $2b_1 + 4b_2 = 0$ is simply twice the first equation. So, that does not give me any extra information so, I have dropped that equation. Now, when you want to solve this notice that b_3 is fixed to be 0.

However, from this equation what you can get is b_1 is $-2b_2$. So, what we have done is instead of b_1 we have put $-2b_2$ retain b_2 and 1. This basically tells us that you can get a null space vector which is $-2 1 0$; however, whatever scalar multiple you use, it will still remain a null space vector.

So, this is easily seen from the following if A times $\beta = 0$ where β is a vector A is a matrix. Let us assume I take some other vector from β which is some $C \beta$, where C is a constant. Then if I plug it back in I will get A times $C \beta = 0$ which because this is scalar I can take it out $C A \beta = 0$. Since, this is 0 C times 0 will be 0. So, this will be also a 0 vector.

So, whenever β is a null space vector then any scalar multiple of that will also be a null space vector that is what is seen by this k here. Nonetheless we have a relationship between these variables which is basically saying $-2x_1 + x_2 = 0$ is a relationship that we can get out of this null space vector.

(Refer Slide Time: 27:24)

Overall summary

- Matrix**
 - Represent data in a matrix form with rows and columns representing samples and attributes respectively
 - Represent coefficients in several equations in a matrix form
- Rank**
 - Number of independent variables or samples
- Nullity**
 - Identifies the number of linear relationships (if any)
- Null Space**
 - Null space vectors provide the linear relationships

Linear Algebra

So, to summarize this lecture as we saw matrix can be used to represent data in rows and columns; representing samples and variables respectively. Matrices can also be used to store coefficients in several equations which can be processed later for further use. The notion of rank, gives you the notion of number of independent variables or samples. The notion of nullity identifies the number of linear relationships, if any between these variables and the null space vectors actually give us the linear relationships between these variables. I hope this lecture was understandable and we will see you again in the next lecture.

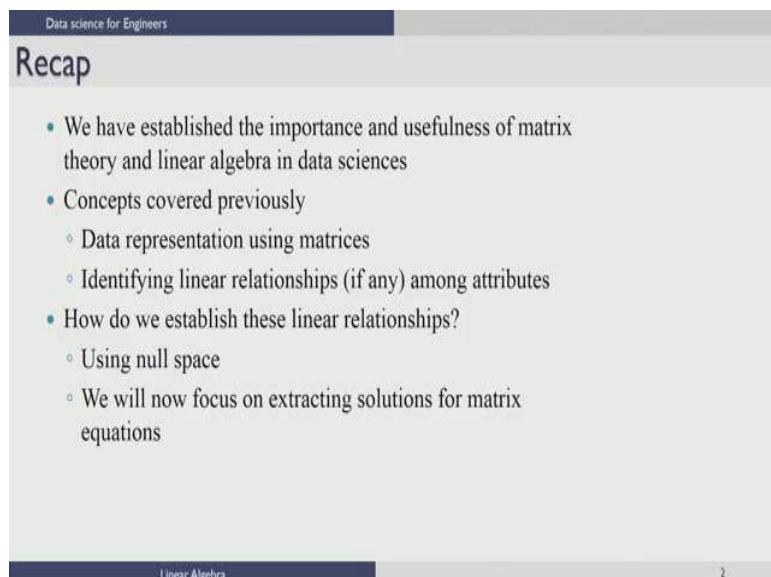
Thank you.

Data Science for Engineers
Prof. Raghunathan Rangaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 13
Solving Linear Equations

In this lecture, we will discuss solutions to linear equation. As we saw in the previous lecture we had established the importance of matrix theory and linear algebra in data science.

(Refer Slide Time: 00:26)



The screenshot shows a presentation slide with a dark blue header bar containing the text "Data science for Engineers". The main content area has a light gray background. At the top left, the word "Recap" is written in a large, bold, black font. Below it is a bulleted list of concepts:

- We have established the importance and usefulness of matrix theory and linear algebra in data sciences
- Concepts covered previously
 - Data representation using matrices
 - Identifying linear relationships (if any) among attributes
- How do we establish these linear relationships?
 - Using null space
 - We will now focus on extracting solutions for matrix equations

At the bottom of the slide, there is a dark blue footer bar with the text "Linear Algebra" on the left and a small number "2" on the right.

The concepts that we covered previously are data presentation using matrices and from matrix of data, we saw how to identify linear relationships if any, among the variables or attributes. We saw that we could establish these linear relationships using the concept of null space.

In this lecture we will focus on the next important topic of extracting solutions for matrix equations, which is the most fundamental aspect of data science, where we might have several equations and we might have to find solutions to those equations. So, we will look at solving matrix equations.

(Refer Slide Time: 01:11)

We consider the following set of equations

$$Ax = b$$

$A(m \times n); x(n \times 1); b(m \times 1)$

- Generalized linear equations can be represented in the above format.
- m and n are the number of equations and variables respectively.
- b is the general RHS commonly used



Linear Algebra

In general when we have a set of linear equations, when we write it in the matrix form, we write this in the form $Ax = b$ where A is generally a matrix of size m by n , and as we saw in the last class m would represent the number of rows and n would represent the number of columns, and for matrix multiplication to work, x has to be of size n by 1 and b has to be of size m by 1.

Now if you take each row of this equation you will have a left hand side and the right hand side. And the left-hand side will have terms corresponding to multiplying the first row of A with x and the right-hand side will have the term corresponding to b . If you take the first row, it will be the first equation and so on. So, from that viewpoint, m represents the number of equations in the system of equations and n represents the number of variables, and in general b is the constant matrix that is used on the right hand side. So, when we write $Ax = b$, this represents a set of m equations in n variables.

(Refer Slide Time: 02:39)

Data science for Engineers

Categorization

- $m = n$**
 - Number of equations and variables are the same
 - Easiest case to solve
- $m > n$**
 - More equations than variables
 - Usually no solution
- $m < n$**
 - Number of equations less than number of variables
 - Usually multiple solutions

We look into these cases independently



Linear Algebra

Now, clearly there are three cases that one needs to address when $m = n$; that means, the number of equations and variables are the same. So, this turns out to be the easiest case to solve. When m is greater than n ; that means, we have more equations than variables. So, we might not have enough variables to satisfy all the equations. In the usual case this will lead to no solution when m is less than n .

The number of equations are less than the number of variables. What this basically means, is that we have lot more variables than necessary to solve the given set of equations. So, in a general case this will usually lead to multiple solutions. So, the first case is the easiest to solve the second case does not have solution usually, and the third case has multiple solutions. What we are going to do, is we are going to look into these cases independently, and then combine all of them using the concept of pseudo inverse.

(Refer Slide Time: 03:43)

Data science for Engineers

Full row and column rank: Concepts

- Consider a matrix data matrix A ($m \times n$)

Full Row Rank <ul style="list-style-type: none"> When all the rows of the matrix are linearly independent Data sampling does not present a linear relationship – samples are independent 	Full Column Rank <ul style="list-style-type: none"> When all the columns of the matrix are linearly independent Attributes are linearly independent
---	--

Row rank = Column rank

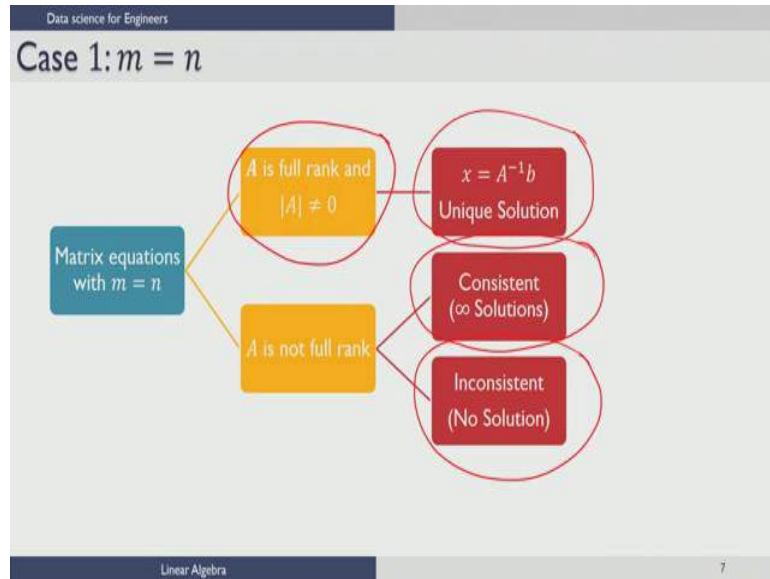
Linear Algebra



We had already discussed the concept of rank in the previous lecture, but let us talk about it again, because this is going to be useful, as we talk about solving equations. If you can consider a matrix A m by n . Now if all the rows of the matrix are linearly independent, then remember we said the rows represent data. So, this, this basically means that the data sampling does not present a linear relationship; that is the samples are independent of each other.

Now when all the columns of the matrix are linearly independent that basically means the variables are linearly independent, because columns represent variable. In a general case if I have a matrix m by n , if m is smaller than n , the maximum rank of the matrix can only be m . So, the maximum rank can be the less of the two numbers. So, in cases where I have A matrix m by n , where n is smaller than m , then the maximum rank that is possible is n . In general whatever be the size of the matrix, it has been established that row rank = column rank. So, you cannot have a different row rank and column rank. What it basically means is, whatever be the size of the matrix, if you have a certain number of independent rows, you will have only those many numbers of independent columns and so on. So, this is something important to bear in mind.

(Refer Slide Time: 05:18)



Now, let us look at the case of $m = n$; that is we have the same number of equations and variables. If A is full rank, what does full rank mean? Now we have the same number of equations and variables $m = n$. So, the rank of the maximum rank of the matrix can be m or n , because both are the same. Now if the rank of the matrix turns out to be m , then it is what is called the full rank matrix, what this basically means, that all of these equations on the left hand side are independent of each other. In other words you can never get any equation on the left hand side as a linear combinations of other equations on the left hand side. In this case there is a unique solution to the $Ax = b$ problem, and that unique solution is $x = A^{-1}b$.

Now from your high school and so on, you would have learned that if the determinant is not 0, A^{-1} is possible to compute. So, one could simply compute $x = A^{-1}b$ as a solution to this problem, the difficulty arises only when A is not full rank; that means, the rank of the matrix is less than n . In this case what it means, is if I take the left-hand side of the equation $Ax = b$, and then make some linear combinations of some rows of A , at least one of the rows of A is going to be a linear combination of the other rows of A ; that is the reason why the rank of the matrix became less than n . In this case, depending on what the values are on the right hand side, you could have two situations; one situation is what we are going to call as a consistent situation.

I will explain this through an example in later slides when you have a consistent situation, then you will have infinite number of solutions. There could be many solutions for $Ax = b$. And in the case where the system of equations become inconsistent, there will be no solution to this problem. So, to summarize when $m = n$ this is what is called a

square matrix, and if the matrix is full rank determinant A not = 0, then there is a unique solution $x = A^{-1} b$, and when A is not full rank there are two situations that are possible! One is what we call as a consistent scenario, where we could have infinite solutions, and the other one is what is called the inconsistent scenario where we might have no solution.

(Refer Slide Time: 08:07)

Data science for Engineers

Case 1: Example 1.1

$$A \mathbf{x} = \mathbf{b} \quad \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 10 \end{bmatrix}$$

R Code

```
A=matrix(c(1,2,3,4),ncol=2, byrow=F)
b=c(7,10)
x=solve(A)%*%b
```

$|A| \neq 0$
 $\text{rank}(A) = 2 = \text{no. of columns}$

- This implies that A is full rank

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 7 \\ 10 \end{bmatrix} = \begin{bmatrix} -2 & 1.5 \\ 1 & -0.5 \end{bmatrix} \begin{bmatrix} 7 \\ 10 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

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

Console output

```
> x
[1] 1
[2] 2
```

Linear Algebra

Let us take a simple example where I have on the top of the screen, the matrix in the form $A x = b$. In this case matrix A is $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ x is $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ 7 10. So, you can notice that there are two equations in two variables x_1 and x_2 . The first equation is basically $x_1 + 3x_2 = 7$ and the second equation is $2x_1 + 4x_2 = 10$. We can see that this is full rank, because whatever multiple of the first column you take, you can never represent the second column and similarly whatever multiple of the first row you take you can never represent the second row, and this can also be seen from the fact that the determinant of A is not 0.

From your high school, you know the determinant is going to be in this case simply 4 times 1, - 2 times 3. So, this is not 0. So, rank of A is 2. So, we have maximum rank that matrix size is 2 by 2, and the rank is 2. So, this implies that the matrix is full rank. Now we said in the previous slide that $x_1 x_2$ can be written as $A^{-1} b$. So, this is A^{-1} inverse matrix $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$, this inverse b. You can do an inverse computation and then calculate the solution as 1 2 and if we put the solution back into the equation, you will see 1 times 2 + 1 times 1 + 3 times 2 is 7 and 2 times 1 + 4 times 2 is 10. So, the solution 1 2 satisfies this equation.

Now if you want to write an r code for this. It is very simple, you write a matrix put the numbers in c and then define the number of columns, you define what b is and then simply use the command solve

for x to get the solution. And as you notice here the solution is 1 2. So, this is a case of full rank where I get an unique solution. The important thing to note here is, no other solution will be able to satisfy these two equations.

(Refer Slide Time: 10:28)

Case 1: Example 1.2

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

$|A| = 0; \text{rank}(A) = 1; \text{nullity} = 1$

- Checking consistency

$$\begin{bmatrix} x_1 + 2x_2 \\ 2x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

$$\text{Row}(2) - 2\text{Row}(1)$$

- The equations are consistent with only one linearly independent equation
- The solution set for (x_1, x_2) is infinite because we have only one linearly independent equation and 2 variables

Now, let us take another example to illustrate what happens when the rows or columns become linearly dependent. So, here is another set of equations and we have to read these equations as $x_1 + 2x_2$ is 5 + $2x_1 + 4x_2$ = 10.

Now if you notice these equations, through the matrix A you will see that, if I multiply the first column by 2 I get the second column. So, the second column is linearly dependent on the first column or the first column is linearly dependent on the second column, whichever way you want to say it. Similarly if you divide for example, the second row by 2 you will get the first row or if you multiply the first row by 2 you get the second row. So, the rows are also linearly dependent and, and as I said before, there is only one independent column and that necessarily means that there will be only one linearly independent row.

Now another way to check this linear dependence is to calculate the determinant of this matrix, and when we calculate the determinant of this matrix, you will get 1 times 4 - 2 times 2, which is 4 - 4 which is 0. So, determinant also says there is linear dependence here. And from the previous lecture we know that when the rank is 1 and the number of columns are 2 the nullity is 1. So, there is one vector in the null space. Now let us look at the equations when you write these equations, as I said before the first equation $x_1 + 2x_2 = 5$, and the second equation is $2x_1 + 4x_2 = 10$.

When we talk about the linear dependence of the rows of A, we are only talking about the left hand side, we never talked about the right hand side. Now whenever the left hand side becomes linearly dependent, if the same linear dependence is maintained on the right hand side also then we have the situation of consistent equations. So, if you take a look at this example, we know the left hand side, if I take the first equation and multiply it by 2 I get the second equation on the left hand side. So, $x_1 + 2x_2$ multiplied by 2 gives me $2x_1 + 4x_2$. Now if the same linear dependence is maintained on the right hand side; that is if I take the first number 5 and multiply it by 2 I should get this number.

In this case, we have constructed this example in such a way that we get this number. Now not only is the left hand side linearly dependent, the same linear dependence is also maintained on the right hand side. So, as a whole the equations become consistent, but linearly dependent on each other. So, in this case you notice that if I solve this equation. I do not have to solve this equation, because I multiply this by 2 I get this equation. So, whatever x_1 and x_2 will solve this equation will also solve the second equation. So, I can drop one of these two equations. Let us assume I drop this equation out. So, I am just left with $x_1 + 2x_2 = 5$, but now notice that I have one equation in two variables, that basically tells me I have one free variable. So, for example, if I take this equation $x_1 + 2x_2 = 5$.

If I set $x_2 = 0$ I get $x_1 = 5$. If I said $x_2 = 1$ I get $x_1 = 3$ and so on. So, all of these are solutions to these equations. I am just pointing out two. Now you can notice that I can take any value for x_2 and then calculate an x_1 , which will satisfy this equation. Since I can take any value for x_2 . There are infinite choices for x_2 corresponding to each one of these infinite choices, I will get a value of x_1 . So, that pair would be a solution to this set of equations. So, when this A becomes rank less than full rank, if the equations are consistent then they will get infinitely number of many infinitely many solutions.

(Refer Slide Time: 14:50)

Data science for Engineers

Case 1: Example 1.3

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$$

$|A| = 0$
 $\text{rank}(A) = 1$
 $\text{nullity} = 1$

- Checking consistency

$$\begin{bmatrix} x_1 + 2x_2 \\ 2x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \end{bmatrix}$$

$$2\text{Row}(1) = 2x_1 + 4x_2 = 10 \neq 9$$

- Thus the equations are inconsistent
- One cannot find a solution to (x_1, x_2)

Linear Algebra

10

Let us take another example to explain the idea of inconsistent equations. Let us look at this example here, this is the same example as the previous case except for one change. This number has been changed from 10 to 9. In the previous case we saw that the left hand side, if I multiply the first equation by

2, I get the second equation. And since in the previous example this number was 10, when I multiplied 5 by 2 I get 10. So, both the equations became consistent and I could drop one equation ; however, if this number was anything other than 10 then what it basically means is, that while the left hand side will be linearly dependent where the second equation will be twice the first equation the right hand side will always be inconsistent.

If this number is not there that basically means whatever x_1 and x_2 values that you use for solving the first equation, If you plug that value into the second equation, your right hand side will always be 10, but you want it to be 9; so which is never possible. So, this is a case where you will not have any solution. So, in other words if this number is anything other than 10, you will have no solution to these equations, this will become a inconsistent case. Now again inconsistent case is possible, only when the rank of the matrix is less than full rank. That is there are some rows which are linearly dependent on the left hand side, which you can again verify by the same determinant 1 times 4 - 2 times 2 is 0. And since the second row is inconsistent, these equations are inconsistent and one cannot find a solution to this problem of $A x = b$.

(Refer Slide Time: 16:44)

Data science for Engineers

Case 2: $m > n$

- This is the case of not enough variables or attributes
- Since the number of equations is greater than the number of variables, in general, not all equations can be satisfied
- Hence it is sometimes termed as a no-solution case
- However, we can identify an appropriate solution by viewing this case from an optimization perspective

Linear Algebra

ii

So, that finishes the case where the number of equations and variables are the same. So, till now we saw the case, where $m = n$. Now let us take the second case, where m is greater than n . Since m is greater than n this basically means that I have more equations than variables. So, this is the case of not enough variables or attributes to solve all the equations. Since the number of equations is greater than the number of variables in general, we will not be able to satisfy all the equations; hence we termed this as a no solution case.

However we still want to identify some solution which makes sense and which we can generalize for all the three cases. So, what we are going to do is, we are going to identify an appropriate solution by viewing this case from an optimization perspective and I explain what that is presently.

(Refer Slide Time: 17:47)

that $(Ax - b)$ is minimized

- Notice that $(Ax - b)$ is a vector
- There will be as many error terms as the number of equations
- Denote $(Ax - b) = e(m \times 1)$; there are m errors $e_i, i = 1:m$
- One could minimize all the errors collectively by minimizing $\sum_{i=1}^m e_i^2$
- This is the same as minimizing $(Ax - b)^T(Ax - b)$

$$(e_1, e_2, e_3, e_4) = e$$
$$a_{11}x_1 + a_{12}x_2 = b_1 (e_1)$$
$$a_{21}x_1 + a_{22}x_2 = b_2 (e_2)$$
$$a_{31}x_1 + a_{32}x_2 = b_3 (e_3)$$
$$\sum_{i=1}^m e_i^2$$

Linear Algebra



Let us look at a solution to $Ax = b$ when the number of equations are more than the number of variables. As we mentioned before we are going to take an optimization perspective here. When we try to solve $Ax = b$, we can write that equation as $Ax - b$, and if there is a perfect solution to the set of equations then $Ax - b$ will be $= 0$. However, since we know that the number of equations are a lot more than the number of variables, there might not be a perfect solution.

So, what we want to do is, we want to identify a solution in such a way that $Ax - b$ is minimized. Why do we want to do this? Notice that $Ax - b$ is a vector and if you take each term in that vector you can think of each of those terms as an error in an equation. So, to give you an example, if I have $a_{11}x_1 + a_{12}x_2 = b_1$ $a_{21}x_1 + a_{22}x_2 = b_2$ $a_{31}x_1 + a_{32}x_2 = b_3$. Now if I had a perfect solution x_1, x_2 which will satisfy all the three equations.

Then when I write this as $a_{11}x_1 + a_{12}x_2 - b_1$, this will be $= 0$; however, when I cannot find a perfect solution then let me call this as an error. In this equation correspondingly I allow an error in this equation, I have error in this equation. So, you notice that there are three errors - as many errors as there are equations. So, how do we collectively minimize all of these errors? One thing to immediately think of both is to minimize $e_1 + e_2 + e_3$, but that would not be a good idea simply, because I could have a_1 as a very large error in the positive direction e_2 as a very large error in the negative direction and e_3 as 0 that will still give me an answer 0. So, that is not a good answer at all, one way to do this is to collectively minimize all of them by minimizing what we call as the sum of squares errors. So, you take this example instead of minimizing $e_1 + e_2 + e_3$. You are going to minimize $e_1^2 + e_2^2 + e_3^2$. In this case notice irrespective of whether e_1, e_2, e_3 are positive or negative as long as they are away from 0. The contribution to the error term will be high. So, it will automatically ensure that you do not go very far away from zero.

Now, this is the least squares solution you could also minimize instead of e_1^2 , you can minimize modulus $e_1 + \text{mod } e_2 + \text{mod } e_3$, because mod is always positive; that is also possible, but in general we are going to talk about least square solution where we minimize this sum of squares of errors. Now this is the same as minimizing $Ax - b$ transpose times $Ax - b$ simply, because $Ax - b$ is e . So, $(Ax - b)^T$ is $e^T e$. So, if I have numbers $e_1 e_2 e_3$ and multiply by $e_1 e_2 e_3$, this will lead to $e_1^2 + e_2^2 + e_3^2$ which is the same as this right here. So, minimizing this, is the same as minimizing $(Ax - b)^T(Ax - b)$.

(Refer Slide Time: 21:38)

Data science for Engineers

Case 2: An optimization perspective

- This optimization problem is
$$\begin{aligned} & \min[(Ax - b)^T(Ax - b)] \\ &= \min[(b^T - x^T A^T)(Ax - b)] \\ &= \min[(x^T A^T Ax - 2b^T Ax + b^T b)] = f(x) \end{aligned}$$
- We observe that the optimization problem is a function of x
- Solving the optimization problem will result in a solution for x
- The solution to this optimization problem is obtained by differentiating $f(x)$ with respect to x and setting the differential to zero

$\nabla f(x) = 0$



Linear Algebra

So, the optimization problem is minimize $(Ax - b)^T(Ax - b)$. After some algebraic manipulation we can write this objective as a function of the solution $f(x)$. We observe that this optimization problem becomes a problem, where the objective is a function of x . Solving this optimization problem will result in a solution for x , which is what we are going for. So, the way to get the solution to deck this optimization problem, is to take $f(x)$ differentiate it with respect to x and set it to 0.

(Refer Slide Time: 22:20)

Data science for Engineers

Case 2: An optimization perspective

- Differentiating $f(x)$ and setting the differential to zero results in
$$2(A^T A)x - 2A^T b = 0$$
$$(A^T A)x = A^T b$$
- Assuming that all the columns are linearly independent
$$\underline{x = (A^T A)^{-1} A^T b}$$



Linear Algebra

So, differentiating $f(x)$ and setting the differential to 0 results in the following equation, and this can be simplified to this equation, where $A^T a$ times x is $A^T b$. Now to solve this equation we will assume that all the columns are linearly independent which allows us to take the inverse of this matrix, and then we can come up with a solution x equal $(A^T A)^{-1} A^T b$. While this solution x might not satisfy all the equations, this solution will ensure that the errors in the equations are collectively minimized. So, this is an optimization view for case 2, where the number of equations are more than the number of variables or m is greater than n .

We will conclude this lecture at this point. What I will do in the next lecture is, take an example to illustrate what happens in case 2 in terms of how you get a solution and whether some of these equations are satisfied or not satisfied and so on, and after that I will move on to case 3 and show an optimization perspective for solving those types of equations, where the number of variables become greater than the number of equations, and then in the next lecture I will also show how all of this can be combined into one elegant solution through the concept of pseudo inverse.

Thank you.

Data Science for Engineers
Prof. Raghunathan Rangaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 14
Solving Linear Equations

We will continue the lecture on solving linear equations. In the last lecture I discussed the case of many more equations and variables, where we might not have a solution and how we can use an optimization perspective to find a solution. In this lecture I am going to give you some examples for that case show you what happens when we apply the solution that we derived last time, and then after that I will go on to look at the case of more variables than equations.

(Refer Slide Time: 00:49)

So, let us look at an $A x = b$ example system as shown in the screen. Here we have a matrix with 3 rows and 2 columns, which basically means that there are 3 equations in 2 variables number of equations more than number of variables. And we have to read these equations as $x_1 = 1$, $2x_1 = -0.5$ and $3x_1 + x_2 = 5$.

So, if you notice these equations you would realize that the first 2 equations are inconsistent. For example, if we were to take the first equation is true then $x_1 = 1$ and if we substitute that value into the second equation you will get $2 = -0.05$. If you were to take the second equation as true then $2x_1$ is -0.5 . So, x_1 will be -0.25 and that would

not solve the first equation. So, these 2 equations are inconsistent. The third equation since it is $3x_1 + x_2$ irrespective of whatever value you get for x_1 you can always use this equation to calculate the value for x_2 ; however, we cannot solve this set of equations.

Now, let us see what is the solution that we get, by using the optimization concept that we described in the last lecture. We said $x = A^T A^{-1} A^T b$. the A matrix is $\begin{bmatrix} 1 & 0 & 2 & 0 & 3 & 1 \end{bmatrix}$. So, A^T matrix is $\begin{bmatrix} 1 & 2 & 3 & 0 \\ 0 & 1 \end{bmatrix}$. Simply plugging in the matrices here.

(Refer Slide Time: 02:30)

Data science for Engineers

Case 2: Example continued

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.2 & -0.6 \\ -0.6 & 2.8 \end{bmatrix} \begin{bmatrix} 15 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

- Thus, the solution for the given example is $(x_1, x_2) = (0, 5)$
- Substituting in the equation shows

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

Linear Algebra

And then doing the calculation gives us this equation which says x_1 x_2 is a matrix times $\begin{bmatrix} 15 \\ 5 \end{bmatrix}$. This is an intermediate step for the calculation. And when you further simplify it you get a solution $x_1 = 0$, $x_2 = 5$. Notice that the optimum solution here that is chosen does not have either one of the 2 cases that we talked about in the last slide, which is $x_1 = 1$ and $x_1 = -0.25$ the optimization approach chooses $x_1 = 0$ and $x_2 = 5$ and when you substitute it back into the equation you get b as $\begin{bmatrix} 0 & 0 & 5 \end{bmatrix}$ whereas, the actual b that we are interested in is $\begin{bmatrix} 1 & -0.5 & 5 \end{bmatrix}$.

So, you can see that while the third equation is being solved exactly the first two equations are not solved for; however, as we described before this is the best solution in a collective minimization of error sense, which is what we defined as minimizing sum of squared of errors. We will now move on to the next example.

(Refer Slide Time: 03:47)

The slide is titled "Case 2: Example". It shows the following system of equations:

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

With notes:

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

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

Linear Algebra

Let us consider another example for us to illustrate something different here. We have taken the same left hand side we have the same a matrix; however, the right-hand side has been modified to be 1 2 5. we have done this for a specific reason which we will see presently. So, when you look at this equation; if you take the first equation it reads as $x_1 = 1$. If you look at the second equation it reads as $2x_1 = 2$. The third equation reads as $3x_1 + x_2 = 5$.

So, from the first equation you can get a solution for $x_1 = 1$ and the second equation since it reads as $2x_1 = 2$, we have to simply substitute the solution that we get from the first equation and see whether the second equation is also satisfied since $x_1 = 1$ 2 times x_1 2 times 1 is 2 the second equation is also satisfied.

Now, let us see what happens to the third equation. The third equation reads as $3x_1 + x_2 = 5$, we already know $x_1 = 1$ satisfies the first 2 equations. So, $3x_1 + x_2 = 5$ would give you $x_2 = 2$. Now you notice that if I get a solution 1 and 2 for x_1 and x_2 ; though the number of equations are more than the variables, the equations are in such a way that I can get a solution for x_1 and x_2 that satisfies all the 3 equations.

Now let us see whether the expression that we had for this case actually uncovers this solution. So, we said $x = (A^T A)^{-1} A^T b$ and we do the same manipulation as the last example except that this b has become 1 2 5 now.

(Refer Slide Time: 05:40)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.2 & -0.6 \\ -0.6 & 2.8 \end{bmatrix} \begin{bmatrix} 20 \\ 5 \end{bmatrix}$$
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

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

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

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

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

Linear Algebra



After some more calculations you will see that $x_1 = 1$ $x_2 = 2$. Thus, the solution is 1 2 and we had already verified that this would solve the equation and we had verified that 1 2 is a solution that we can directly get by observation from the previous slide.

So, the important point here is that if we have more equations than variables then you can always use this least square solution which is $(A^T A)^{-1} A^T b$. The only thing to keep in mind is that $(A^T A)^{-1}$ exists if the columns of A are linearly independent. If the columns of A are not linearly independent, then we have to do something else which you will see as we go through this lecture.

(Refer Slide Time: 06:32)

Data science for Engineers

Case 3: $m < n$

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

Linear Algebra



So, that finishes the case where the number of equations are more than the number of variables. Now let us address the last case where the number of equations are less than the number of variables, which would be m less than n in this case we address the problem of more attributes or variables than equations.

Now since I have many more variables and equations I would have infinite number of solutions the way to think about this is the following. If I had, let us say, 2 equations and 3 variables. You can think of this situation as one where you could choose any value for x_3 and then simply put it into the 2 equations. And whatever are the terms with respect to x_3 you collect them and take them to the right-hand side; that would leave you with 2 equations and 2 variables and once we solve for that 2 equations and 2 variables we will get values for x_1 and x_2 .

So, basically what this means is that, I can choose any value for x_3 and then corresponding to that I will get values for x_1 and x_2 . So, I will get infinite number of solutions. Since I have infinite number of solutions then the question that I ask is how do I find one single solution from the set of infinite possible solutions? Clearly if you are looking at only solvability of the equation, there is no way to distinguish between this infinite possible solutions. So, we need to bring some other metric that we could possibly use, which would have some value for us to pick one solution that we can say is a solution to this case.

(Refer Slide Time: 08:12)

Data science for Engineers

Case 3: An optimization perspective

- Pose the following optimization problem
$$\min\left(\frac{1}{2}x^T x\right) \text{ s.t. } Ax = b$$
- Define a Lagrangian function $f(x, \lambda)$
$$\min\left[f(x, \lambda) = \frac{1}{2}x^T x + \lambda^T(Ax - b)\right]$$
- Differentiating the Lagrangian with respect to x , and setting to zero

$$x + A^T \lambda = 0$$


Linear Algebra

Similar to the previous example we are going to take an optimization view here, what we are going to do is we are going to minimize $x^T x$, this half is just to make sure the solution comes out in a

nice form. And notice here something that is important we also have a constraint for this optimization problem s dot t dot means subject to.

So, I want to minimize this half; $x^T x$ subject to the constraint $A x = b$. So, in other words what we are saying is whatever solution we get for x that has to necessarily satisfy this equation. And this is not a problem we can find infinite number of solutions x which will satisfy these equations. So, what this objective does is of all of those solutions how do I pick, that one solution which will minimize this $x^T x$. We have to think about a rationale for, why we would choose x transpose x as an objective.

This basically says that of all the solutions I want the solution which is closest to the origin is what this is saying in terms of x transpose x . From an engineering viewpoint one could justify this as the following; if you have lots of design parameters that you are trying to optimize and so on, you would like to keep the sizes small for example, so you might want small numbers. So, you want to be as close to origin as possible this is just one justification for doing something like this nonetheless this is one way of picking one solution from this infinite number of solutions.

Now, in the previous example and in this example, we are solving these optimization problems; however, we have not taught in this course how to solve optimization problems. For people who already know how to solve optimization problems this would be obvious. For other participants who do not know how to solve optimization problems, I would encourage you to just bear with me and then go through this solution and see what the solution form is and once this module on linear algebra is finished we will have a couple of modules on optimization from the viewpoint of data science.

So, when we do that, you will see how we solve these kinds of optimization problems. The optimization problem that we solved for the last case is what is called an unconstrained optimization problem because there are no constraints to that problem whereas, this problem that we are solving is called a constrained optimization problem because while we have an objective we also have a set of constraints that we need to solve.

So, you will have to bear with us till you go through the optimization module to understand this. Interestingly it is generally a good idea to teach linear algebra on optimization, but interestingly, some of the linear algebra concepts you can view as optimization problems and solving optimization problems requires lots of linear algebra concepts. So, in that sense they are both coupled. In any case to solve optimization problems of this form we can define what is called a Lagrangian function $f(x)$ comma λ , λ are extra parameters that we introduce into this optimization formulation. And what you do is you

minimize this Lagrangian with respect to x to get a set of equations. And you also minimize this with respect to Lagrangian which will back out the constraint. So, whatever solution you have, has to solve both the differentiation with respect to x which should give you $x + A^T \lambda = 0$ and also differentiation with λ which will simply give you $A x - b = 0$. That would basically say that whatever solution you get, that has to satisfy the equation $A x = b$. We will see how this is useful in identifying a solution.

(Refer Slide Time: 12:35)

Data science for Engineers

Case 3: An optimization perspective

$$x = -A^T\lambda$$

Pre-multiplying by A

$$Ax = b = -AA^T\lambda$$

Thus we obtain $\lambda = -(AA^T)^{-1}b$ assuming that all the rows are linearly independent

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

Linear Algebra

So, let us look at this equation $x + A^T \lambda = 0$. So, from this we can get a solution for x which is $-A^T \lambda$. Now what you could do is; you do not know x and you do not know λ also. So, there has to be some way of finding out both of them. So, what we are going to do is we are going to use the knowledge that any solution that we get has to satisfy the equation $A x = b$.

So, what we are going to do is we are going to pre-multiply this x by A . So, we premultiply on both sides so, we get a $x = -A^T \lambda$ by premultiplying this equation by A . Now since any solution x satisfies $A x = b$, I can replace this $A x$ by b and I get this equation $b = -A A^T \lambda$ and from this equation we can get λ to be $-A A^T$ inverse b . And this is possible and this inverse exists only if all the rows are linearly independent.

Now, since we have an expression for λ we can substitute that expression here and we will get $x = -A^T \lambda$ and your λ is this expression which is from here. So, this solves for x in the equation $A x = b$. And since we use this idea here the x that we get is such that $A x = b$ that is satisfies the original equation.

(Refer Slide Time: 14:17)

Data science for Engineers

Case 3: Example

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

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

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

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


Linear Algebra

Now, let us take an example to understand this. I have an $A x = b$ here, I have a as $1 \ 2 \ 3 \ 0 \ 0 \ 1$ and b as $2 \ 1$. So, again notice here since there are 2 equations, I have 2 rows and 3 equation 3 variables, I have 3 columns and these equations are read as $x_1 + 2x_2 + 3x_3$ is 2 and $x_3 = 1$. Now clearly when you look at this equation you will notice that $x_3 = 1$ has to be a solution. So, the question is how do I choose x_1 and x_2 , nonetheless we will use the optimization solution to actually see what happens here.

So, the optimization solution from the previous slide is the following $x = A^T (A A^T)^{-1} b$. Now A^T is $1 \ 2 \ 3 \ 0 \ 0 \ 1$ here. And this is my A and A^T again I take an inverse of this and b now is $2 \ 1$.

(Refer Slide Time: 15:28)

Data science for Engineers

Case 3: Example

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

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

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

R Code

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

Console output

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



Linear Algebra

And when I do some more algebra I finally get a solution to x_1 x_2 x_3 which is the following; And we had already seen that $x_3 = 1$ has to be a solution because the last equation basically said $x_3 = 1$. Now x_1 and x_2 you could have found several numbers to satisfy the first equation after you choose $x_3 = 1$ of all of these this solution says this - 0.2 - 0.4 is the minimum norm solution or this vector is the closest vector from the origin; that satisfies my equation $A x = b$. So, I can finally, say my solution x_1 x_2 x_3 is - 0.2 - 0.41.

(Refer Slide Time: 16:14)

Data science for Engineers

Case 3: Example

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

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

And you can easily verify that this satisfies the original equation since x_3 is 1, the second equation is $x_3 = 1$.

So, that gets satisfied when you look at the other equation you have one times - 0.2 + 2 times - 0.4. That will be - 0.2 - 0.81 + 3 times 1 will give you 3 - 1 = 2 which is this. So, the solution that we found satisfies the original equation and this also turns out to be the minimum norm solution as we discussed.

(Refer Slide Time: 16:56)

Data science for Engineers

Generalization

- The described cases cover all the scenarios one might encounter while solving linear equations
- Is there any form in which the results obtained for cases 1, 2 and 3 can be generalized?
- The concept we used to generalize the solutions is called as Moore-Penrose pseudo-inverse of a matrix
- The pseudo inverse is used as follows
$$Ax = b$$

The solution becomes
$$x = A^{-1}b$$
- Singular Value Decomposition can be used to calculate the pseudo inverse or the generalized inverse (A^+)

Linear Algebra

So, when we have a set of linear equations we basically said that there are 3 cases that one needs look at, one case is where number of equations and variables are the same $m = n$. The second case is where the number of equations are lot more than the number of variables m greater than n . And the third case was when number of equations less than number of variables m less than n . And we saw that one case is an exact solution if it is a full rank matrix.

And if it is not a full rank matrix then you could have infinite solutions or no solutions, and interestingly the next 2 cases covers these 2 aspects when I have lot more equations than variables I have a no solution case, and when I have lot more variables than equations I have infinite solution case and since we are able to solve all the 3 we should be able to use the solution to the case 2 and 3 for the case one where the rank is not full. And depending on whether it is a consistent set of equation or inconsistent set of equation you should be able to use the corresponding infinite number of solutions or no solutions result right?

So, in some sense we understand that there should be some generalization of all of these results. So, that we can write one equation which solves all of these cases square rectangular cases and so on. So, that is a question that we are asking, is there any form in which the results obtained from cases 1, 2 and 3 can be generalized. It turns out that there is a concept that we can use to generalize all of these, this is what is called the Moore Penrose pseudo inverse of a matrix.

So, when we typically have equations of the form $a x = b$, we write $x = A^{-1}b$ as a solution. The generalization of this is to write x is $A + I$ have used this term to denote the pseudo inverse b . And as long as we can calculate the pseudo inverse in a fashion that irrespective of the size of A , irrespective of whether the columns and rows are dependent or independent.

If I can write one general solution like this which will reduce to the cases that we discussed in this lecture, then that is a very convenient way of representing all kinds of solutions instead of looking at whether the number of rows are more, number of columns are more, is rank full and so on. All of them if they can be subsumed in one expression like this it would be very nice and it turns out that there is an expression like that and that expression is called the pseudo inverse.

Now, the pseudo inverse a for a can be calculated using a singular value decomposition as one technique. There are many other ways of computing this, but singular value decomposition is one way of computing this. And as far as this course is concerned you just need to know that we can compute this. We do not have to really worry about how singular value decomposition is done.

(Refer Slide Time: 20:17)

Data science for Engineers

Two examples revisited

Example 2 <pre>R Code A=matrix(c(1,2,3,0,0,1),ncol=2, byrow=F) b=matrix(c(1,2,5),ncol=1, byrow=F) library(MASS) x=ginv(A) %*% b</pre> <p>Solution</p> <pre>> x [1] 1 [2] 2</pre>	Example 3 <pre>R Code A=matrix(c(1,0,2,0,3,1),ncol=3, byrow=F) b=c(2,1) library(MASS) x=ginv(A) %*% b</pre> <p>Solution</p> <pre>> x [1] -0.2 [2] -0.4 [3] 1.0</pre>
---	---

Linear Algebra

So, how do I get this in R? So the way you do this in R is you use this library and the pseudo inverse is usually calculated using this generalized inverse a . Here g stands for generalized. So, what R does is whatever size of the problem you give here we have given 2 different examples,

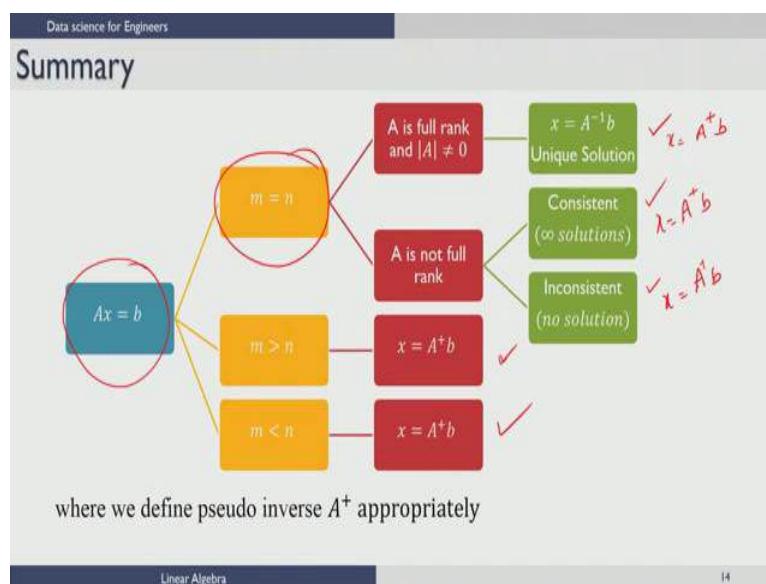
where one example has more equations than variables the second example has more variables than equation.

These are the examples that were picked from this lecture itself and we show that irrespective of whatever be the sizes of this matrices A and b , we use the same equation g inverse A and the solution 1 2 that we got in one example and the solution - 0.2 - 0.4 and one we got in the other case come out of this g inverse.

Now, the key point to understand is you simply use g inverse in R to get these solutions, but the interpretation of these solutions is what we have taught in this class. So, interpretation for this solution here is that; this is the least square solution or this is the solution that will minimize the errors collectively or this is a solution that will minimize $e1^2 + e2^2$ and so on.

This is what is called the minimum norm solution. While there are infinite number of solutions this is a solution that is the closest to origin. So, that is the interpretation for these 2 solutions that that we want to keep in mind as far as solving linear equations is concerned, nonetheless the operationalization for how to use R is very simple you simply use g inverse as a function.

(Refer Slide Time: 22:21)



So, let me summarize this lecture, we said we are interested in solving equations of the form $A x = b$. We talked about 3 cases $m = n$ and $m > n$ if A is full rank unique solution $A^{-1}b$. If A is not full rank there are 2 possibilities either the equations are consistent or inconsistent. And if m is greater than n we look at a least square solution and if m is less than n then we look at a least norm solution.

We can write this as $A^{-1}b$ or I could also write this as pseudo inverse b . In this case the pseudo inverse and A inverse will be exactly the same and as I mentioned before since these 2 cases are covered by these 2. I should be able to use the same a pseudo inverse b for both these cases also without worrying about whether they are consistent inconsistent and so on. In all of these cases I will get a solution by using the idea of generalized inverse.

So, this concludes the section on solving linear equations irrespective of whether it is a square or a rectangular system or not, worrying about really whether the columns are dependent independent and so on. You can use generalized inverse as one unifying concept to find a solution to all these cases.

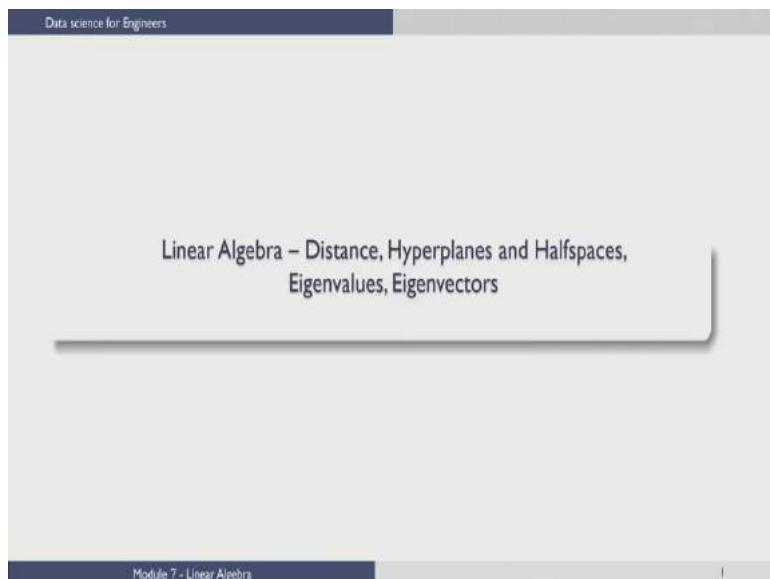
Thank you and in the next lecture we will take a geometric view of the same equations and variables that is useful in data science.

Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 15

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors

(Refer Slide Time: 00:12)



In the previous lectures we looked at linear algebra. But we took a linear algebraic view where we looked at equations and variables and solvability of these equations and so on. The same subject we could also take a geometric view, where we think about vectors and hyperplanes, half spaces and so on. So, we are going to cover that in the next couple of lectures that we are going to have on linear algebra. While we do this, we are going to cover the ideas of distance, hyperplanes, half spaces, Eigenvalues Eigenvectors. Now, some of these are things that would be very well known to most of you nonetheless, for the sake of completeness, I will go through all of these ideas and then I will use all of those ideas, when we describe hyper planes, half spaces and so on.

(Refer Slide Time: 01:21)

Data science for Engineers

Review

- So far we have discussed linear algebra and matrix theory from a data science perspective
- We will provide some geometric interpretations now
- This section covers the following
 - Vectors
 - Notion of distance
 - Projections
 - Hyperplanes
 - Halfspaces
 - Eigenvalues and eigenvectors

Module 7 - Linear Algebra 2

So, we will cover vectors notion of distance, we will talk about projections, we will talk about hyper planes, we will talk about half spaces and then we will talk about Eigenvalues and eigenvectors in this lecture. Till now, if we have been looking at a $X = b$ and X as set of variables that needs to be calculated. So, we have been using this notation $x_1 x_2$ as a vector, where we have been interpreting this as a solution to a variable x_1 and a solution to variable x_2 and so on.

(Refer Slide Time: 01:37)

Data science for Engineers

Vectors and lengths

- Consider

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- X is a data point in a 2 dimensional plane with x_1 and x_2 as the distances along the X_1 and X_2 axes respectively.
- X can also be considered as a vector between the origin and the data point
- The length (magnitude) of this vector is

$$d = \sqrt{x_1^2 + x_2^2}$$

Another way to think about the same vector X is to think of this as actually a point in a 2-dimensional space and here we say, it is a 2 dimensional space because there are 2 variables. So, for example, if you take x_1 and x_2 you could think of this, as being a point in a 2 dimensional space, where there is one axis that represents x_1 and there

is another axis that represents x_2 , and depending on the value of the an x_1 and x_2 you will have a point anywhere in this plane.

So, for example, if you have let us say 1 as your vector, and if this is one and this is one, then the point will be here and so on. So, what we are doing here is, we are looking at vectors as points in a particular dimensional space. Since, there are 2 numbers here we talked about 2-dimensional space if for example, there are 3 numbers here, then it would be a point in a 3 dimensional space, you could also think of this as a vector and we define the vector from the origin.

So, I could think of this X as a vector, where I connect origin to the point. So, this is another view of the same vector X and once we think of this as a vector then, vector has both direction and magnitude. So, in this case the direction is this and the magnitude is, what we think of as a distance from the origin and in this case we all know, this well-known formula for Euclidean distance, which is root of $(x_1^2 + x_2^2)$ right? So, that is the distance of this point from the origin.

(Refer Slide Time: 03:59)

Data science for Engineers

Vectors and lengths: Example

- Consider the point $A = (3,4)$ in a two dimensional plane

$$A = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$
$$d = \sqrt{3^2 + 4^2} = 5 \text{ units}$$

- Important: Geometric concepts are easier to visualize in 2D or 3D
- Difficult to do so in the higher dimensions
- However, the fundamental mathematics remain the same irrespective of the dimension of the vector

Module 7 - Linear Algebra

Now, just as a very, very simple example, if you have a 0.34 then you can find the distance from the origin is root of $(3^2 + 4^2)$ is going to be = 5. It is important to notice that the geometric concepts are easier to visualize in 2D or 3D; however, they are difficult to do. So, in higher dimensions, nonetheless since the fundamental mathematics remain the same what we can do is, we can understand these basic concepts using 2D and 3D geometry and then simply scale the number of dimensions, and then most of the things that we understand and learn will be the same at higher dimensions also.

(Refer Slide Time: 04:45)

Data science for Engineers

Vectors and distances

- Consider another example with two points X^1 and X^2

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

- The distance between these two can be calculated

$$l = |X^2 - X^1|_2$$

$$l = \sqrt{(x_1^2 - x_1^1)^2 + (x_2^2 - x_2^1)^2}$$

$$l = \sqrt{(X_2 - X_1)^T (X_2 - X_1)}$$

Module 7 - Linear Algebra

So, in the previous slide we saw one point in 2 dimensions. Now, let us consider a case where we have 2 points in 2 dimensions. We have x_1 here, which has 2 numbers representing the 2 coordinates and we have x_2 here, which also represents the 2 coordinates. Now, we ask the question as to, whether we can define a vector which goes from x_1 to x_2 . So, pictorially this is the way in which, we are going to define this vector. What we do is, we draw a line starting from x_1 to x_2 and this vector is $x_2 - x_1$, the direction of the vector is given by this here, much like the previous case every vector will have a direction and a magnitude.

So, we might ask what is the magnitude of this vector and that is given by the wellknown formula that we see right here. Where what you do basically is, you take the x_1 coordinate of this point and this point take the difference square it, take the x_2 coordinate of this point the x_2 coordinate of this point, take the difference and square it add both of them and take a root and that is the equation that we have here.

This is the length of this vector right here, this also can be written in a compact form as given here, which is root of $(x_2 - x_1)^T (x_2 - x_1)$.

(Refer Slide Time: 06:30)

Data science for Engineers

Vectors and distances: Example

- What is the distance between points A and B , where A is $\begin{pmatrix} 2 \\ 7 \end{pmatrix}$ and B is $\begin{pmatrix} 5 \\ 3 \end{pmatrix}$
- Using the concept of distance introduced before

$$A = \begin{pmatrix} 2 \\ 7 \end{pmatrix}, B = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$
$$l = \sqrt{(5 - 2)^2 + (3 - 7)^2}$$
$$l = 5 \text{ units}$$


Module 7 - Linear Algebra

Two simple examples to illustrate this, if I have 2 points A and B where A is 2 7 b is 5 3 then, the distances you take the difference between 5 and 2 and then square it and then, take the difference between 3 and 7 and then square it and then you will get your length as 5. So, that would be the length of the line that is, drawn between the 2 points A and B.

(Refer Slide Time: 06:59)

Data science for Engineers

Unit vector

- A unit vector is a vector with magnitude 1 (distance from origin)
- Unit vectors are used to define directions in a coordinate system
- Any vector can be written as a product of a unit vector and a scalar magnitude

$$\underline{A} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} \checkmark \quad \underline{A} = 5\hat{a}$$

Magnitude of A : $|A| = \sqrt{3^2 + 4^2} = 5$ ✓

$$\hat{a} = \frac{\underline{A}}{|A|} = \begin{pmatrix} 3/5 \\ 4/5 \end{pmatrix}$$


Module 7 - Linear Algebra

Now, it is useful to define vectors with unit length, because once you write a vector in unit length any other vector in that direction, can be simply written as the unit vector times the magnitude of the vector that you are interested in.

So, how do I define a unit vector, let us take this vector $\begin{pmatrix} 3 \\ 4 \end{pmatrix}$, we know that the distance from the origin for this vector is root of $(3^2 + 4^2) = 5$. So, to define a unit vector what you do is, you take the vector and divide it by the magnitude of the vector. So, in this case it is 5. So, the unit vector becomes $\begin{pmatrix} 3 \\ 4 \end{pmatrix} / 5$. So, the interesting thing is that, this unit vector is in the same direction as $\begin{pmatrix} 3 \\ 4 \end{pmatrix}$; however, it has magnitude 1. So, I could write $\begin{pmatrix} 3 \\ 4 \end{pmatrix}$ itself as 5 times $\begin{pmatrix} 3 \\ 4 \end{pmatrix} / 5$. So now what has happened is this is a unit vector and this $\begin{pmatrix} 3 \\ 4 \end{pmatrix}$ has magnitude 5, which is what we derived here.

(Refer Slide Time: 08:04)

Data science for Engineers

Orthogonal vectors

- Two vectors are orthogonal to each other when their dot product is 0
- Dot product (scalar product) of two n dimensional vectors A and B

$$A \cdot B = \sum_{i=1}^n a_i b_i$$

- Thus the vectors A and B are orthogonal to each other if and only if

$$A \cdot B = \sum_{i=1}^n a_i b_i = A^T B = 0$$

Module 7 - Linear Algebra

We introduce the next concept, which is important for us to understand many of the things that we are going to teach. If there are 2 vectors, we call these vectors as orthogonal to each other when their dot product is 0. So, how do I define the dot product? So, if I take 2 vectors A and B A dot B is simply $\sum_{i=1}^n a_i b_i$.

So, basically what you do is, if you have 2-dimensional vector then you take the 2 x coordinates multiply them, and then you take the 2 y coordinates and multiply them and add both of them you will get the dot product. This dot product again much like the distance that we saw before, can also be written in a compact form as $A^T B$ you can quite easily see that this and this will be the same, and if this dot product turns out to be 0 then we call this vectors A and B as being orthogonal to each other.

(Refer Slide Time: 09:06)

Data science for Engineers

Orthogonal vectors: Example

- Consider the vectors v_1 and v_2 in 3D space. Identify if they are orthogonal to each other

$$v_1 = \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix}$$
$$v_2 = \begin{bmatrix} 2 \\ 5 \\ 2 \end{bmatrix}$$

R Code

```
v1=c(1,-2,4)
v2=c(2,5,2)
N=t(v1)%*%v2
```

Console Output

```
> N
[1] 0
```

- Taking the dot product of the vectors

$$\underline{v_1} \cdot \underline{v_2} = \underline{v_1^T} \underline{v_2} = [1 \ 2 \ 4] \begin{bmatrix} 2 \\ 5 \\ 2 \end{bmatrix} = 0$$

- Hence, the vectors are orthogonal

Module 7 - Linear Algebra

9

So, let us take an example to understand this, let us take 2 vectors in 3-dimensional space. Let us say, I have one vector which is 1 - 2 4 and I have the other vector which is 2 5 2 and if I take a dot product between these 2, which is $v_1^T v_2$ or $v_2^T v_1$, both will be the same. I have v_1^T which is 1 - 2 4 and this is 2 5 2, if this will be one times 2 - 5 times 2 + 4 times 2 you will see that goes to 0. So, we say that these 2 vectors are orthogonal to each other.

(Refer Slide Time: 09:40)

Data science for Engineers

Orthonormal vectors

- Orthonormal vectors are orthogonal vectors with unit magnitude
- Example

$$v_1 = \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix} / \sqrt{1^2 + (-2)^2 + 4^2}$$
$$v_2 = \begin{bmatrix} 2 \\ 5 \\ 2 \end{bmatrix} / \sqrt{2^2 + 5^2 + 2^2}$$

Unit vectors

- Note that we have taken the vectors from the previous example and converted them into unit vectors by dividing them with their magnitudes.
- All orthonormal vectors are orthogonal



Module 7 - Linear Algebra

Now, take the same 2 vectors, which are orthogonal to each other and you know that, when I take a dot product between these 2 vectors it is going to go to 0. If I also impose the condition, that I want each of these vectors to have unit magnitude then what I could possibly do? Is I could take this vector and then divide this vector by the magnitude of this vector.

So, this is going to be root of one squared + - 2 whole squared + 4 squared. Similarly, I can take this vector and divide this vector by the magnitude of the same vector, which is going to be root of 2 squared + 5 squared + 2 squared. Now, these 2 are unit vectors, because the magnitudes are the same and these unit vectors also turn out to be orthogonal to each other, the orthogonal property is not going to be lost, because these are scalar constants. So, while you take $v_1^T v_2$ or $v_2^T v_1$, it will still turn out to be 0. So, these vectors will still be orthogonal to each other. However now individually, they also have unit magnitude such vectors are called are orthonormal vectors, that we have defined here. Notice that all orthonormal vectors are orthogonal by definition.

(Refer Slide Time: 11:14)

R^2

Basis vectors

Let us consider two vectors $v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2v_1 + 1v_2$$

$$\begin{bmatrix} 4 \\ 4 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 4v_1 + 4v_2$$

$$\begin{bmatrix} 1 \\ 3 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1v_1 + 3v_2$$

v_1 and v_2 are the basis vectors for R^2

Module 7 - Linear Algebra

Now, we are going to come to the next interesting concept that we would need in data science quite a bit and I am going to explain this concept through very, very simple examples. This can also be very formally defined, what I am going to do is, I am going to try and explain this in a very simple fashion. So that you understand what this means and I also want to give a context, in terms of why these are some things that we are interested in looking at from a data science viewpoint.

So, we are going to introduce the notion of basis vectors. So, the idea here is the following, let us take R squared which basically means that, we are looking at vectors in 2 dimensions. So, I could come up with many many vectors, right? So, there will be infinite number of vectors, which will be in 2 dimensions. So, this is like saying, if I take a 2-dimensional space how many points can I get? So, I can get infinite number of points. Which is what has been represented here.

So, I have put in some vectors and then these dots represent that, there are infinite number of such vectors in this space. Now, we might be interested in understanding, something more general than just saying that there are infinite number of vectors here. So, what we are interested in is, if we can represent all of these vectors using some basic elements and then some combination of these basic elements, is what we are interested in.

Now, let us consider 2 vectors for example, $v_1 = 1 \ 0$ and $v_2 = 0 \ 1$. Now, if you take any vector that I have here, let us say take $2 \ 1$, I can write $2 \ 1$ as some linear combination, of this vector + this vector. Similarly, take $4 \ 4$, I can write $4 \ 4$ as a linear combination of this vector + this vector and that would be true for any vector that you have in this space.

So, in some sense what we say is that, these 2 vectors characterize the space or they form a basis for the space and any vector in this space can simply be written as a linear combination of these 2 vectors. Now you notice, the linear combinations are actually the numbers themselves. So, for example, if I want this to be written as a linear combination of $1 \ 0 \ 1 \ 0 \ 1$, the linear combination the scalar multiples are 2 which is this, and 1 which is this similarly 4 here 4 here and so on.

So, the key point being, while we have infinite number of vectors here, they can all be generated as a linear combination of just 2 vectors and we have shown here, these 2 vectors as $1 \ 0 \ 1 \ 0 \ 1$. Now, these 2 vectors are called the basis for the whole space, if I can write every vector in the space as a linear combination of these vectors and these vectors are independent of each of them.

Then we call them as a basis for the space. So, why do you want these vectors to be independent of each other? We want these vectors to be independent of each other, because we want every vector, that is in the basis to generate unique information. If they become dependent on each other, then this vector is not going to bring in anything unique. So, basis has 2 properties, every vector in the basis should some bring something unique, and these vectors in the basis should be enough, to characterize the whole space, in other words the vector should be complete.

(Refer Slide Time: 15:07)

Data science for Engineers

Basis vectors

- Basis vectors are set of vectors that are independent and span the space
- Example:
 - Two vectors $v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
 - Can span R^2 and are independent and hence form the basis for the R^2 space.



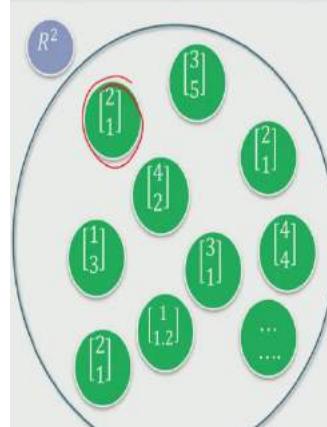
Module 7 - Linear Algebra

So, this we can formally say as the following, basis vectors for any space are a set of vectors that are independent and span the space and the word span basically means that, any vector in that space, I can write as a linear combination of the basis vectors. So, the previous example, we saw that the 2 vectors $v_1 1 0$ and $v_2 0 1$, can span the whole R squared and you can clearly see that they are independent of each other, because no multiple scalar multiple of this will be able to give you this vector .

(Refer Slide Time: 15:49)

Data science for Engineers

Basis vectors are not unique



Consider two vectors $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$$\begin{bmatrix} 2 \\ 1 \end{bmatrix} = 1.5 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (-0.5) \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 1.5v_1 + (-0.5)v_2$$
$$\begin{bmatrix} 4 \\ 4 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 0 \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 4v_1 + 0v_2$$
$$\begin{bmatrix} 1 \\ 3 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + (-1) \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 2v_1 + (-1)v_2$$

Hence, this v_1 and v_2 are also basis vectors for R^2

Module 7 - Linear Algebra

So, the next question that immediately pops up in ones head is, if I have a basis vector, are they unique? Now it turns out these basis

vectors are not unique, you can find many many sets of a basis vectors, all of which would be equivalent. The only conditions are that they have to be independent and should span the space. So, take the same example and let us consider 2 other vectors, which are independent.

So, the same example as before, where we had used 2 basis vectors $1\ 0$ and $0\ 1$, I am going to replace them by $1\ 1$ and $1\ -1$. Now, the first thing that we have to check is, if these vectors are linearly independent or not and that is very easy to verify. If I multiply this vector by any scalar, I will never be able to get this vector. So, for example, if I multiply this by -1 I will get -1 and -1 , but not $1\ -1$. So, these 2 are linearly independent of each other.

Now, let us take the same vectors and then see what happens. So, remember we represented $2\ 1$ in the previous case, as 2 times $1\ 0 + 1$ times $0\ 1$. Now, let us see whether I can represent this $2\ 1$ as a linear combination of $1\ 1$ and $1\ -1$. So, if you look at this, this is the linear combination notice; however, because of the way I have chosen these vectors, these numbers are not the same as this.

So, in the previous case when we use $1\ 0$ on $0\ 1$, we said this can be written as 2 times $1\ 0 + 1$ times $0\ 1$; however, the numbers have changed now, nonetheless I can write this as a linear combination of these 2 basis vectors.

Let us take this $4\ 4$ as an example. So, that can be written as an interesting linear combination, which is 4 times $1\ 1 + 0$ times $1\ -1$ right? So, that will give you $4\ 4$ similarly $1\ 3$ can be written as, 2 times $1\ 1 + -1$ times $1\ -1$. So, this is another linear combination of the same basis vectors.

So, the key point that I want to make here is that, the basis vectors are not unique there are many ways in which you can define the basis vectors; however, they all share the same property that, if I have a set of vectors which I call as a basis vector, those vectors have to be independent of each other and they should span the whole space and whether you take $0\ 1\ 1\ 0$ and call it a basis set or you take $1\ 1$ and $1\ -1$ and call the basis set, both are all right and you can see that, in each case the vectors are independent of each other and they span the whole space.

An interesting thing to note here though is that, I cannot have 2 basis sets which have different number of vectors, what I mean here is in the previous example though the basis vectors were $1\ 0$ and $0\ 1$, there were only 2 vectors. Similarly, in this case the basis vectors are $1\ 1$ and $1\ -1$.

However, there are still only 2 vectors. So, while you could have many sets of basis vectors, all of them being equivalent, the number of

vectors in each set will be the same. They cannot be different and this is easy to see. I am not going to formally show this, but this is something that you should keep in mind, in other words for the same space you cannot have 2 basis sets - one with n, vectors other one with m vectors - that is not possible. So, if it is a basis set for the same space, the number of vectors in each set should be the same. Now, I do not want you to think that the basis set will always have to be the number of elements in the vector.

(Refer Slide Time: 19:54)

Data science for Engineers

Basis vectors

Consider two vectors $v_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \end{bmatrix}$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + 0 \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \end{bmatrix} = 1v_1 + 0v_2$$

$$\begin{bmatrix} 7 \\ 7 \\ 11 \\ 15 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} + 1 \begin{bmatrix} 4 \\ 1 \\ 2 \\ 3 \end{bmatrix} = 3v_1 + 1v_2$$

So, to give you another example, we have generated this data in a particular fashion. Consider now this set of vectors right? There are infinite number of vectors here and we will say all of these vectors are in space R^4 , which basically means that there are 4 components in each of these vectors.

Now, what we want to ask is, what is the basis set for these kinds of vectors? Now when I do this here, the assumption is the extra vectors that I keep generating, the infinite number of them, all follow certain pattern that these vectors are also following and we will see what that pattern is. So, what we can do is, we can take, let us say 2 vectors here, in this case this is how this example has been constructed to illustrate an important idea. Let us take this vectors v_1 which is $1 \ 2 \ 3 \ 4$ and let us take some vector here, in this set let us take this vector here, and then see what happens, when I try to write it as a linear combination of these 2 vectors.

So, I can see that if I take this I can write it as 1 times this + 0 times the second vector. So, that is one linear combination, now let us take some other vector here. So, let us say for example, we have taken this vector $7 \ 7 \ 11 \ 15$, we can see that that can be written as a linear

combination of 3 times the first vector + 1 times the second vector and so on.

Now, you could do this exercise for each one of these vectors and you will be able to see, because of the way we have constructed these vectors, you will be able to see that each one of these vectors, I can write as a linear combination of v_1 and v_2 . So, what this basically says is the following, it says that, though I have 4 components in each of these vectors, that is, all of these vectors are in R^4 , because of the way in which these vectors have been generated, they do not need 4 basis vectors to explain them, all of these vectors have been derived as a linear combination of just 2 basis vectors, which are given here and here.

So, in other words all of these vectors would occupy A_2 dimensional, what we call as a subspace in R^4 right? So, if you take every vector in R^4 , without leaving out anything then, you would need 4 basis vectors to explain all of them. However, these vectors have been picked in such a way, that they are only linear combination of these 2 vectors. So, I just need 2 vectors to represent all of this. So, I say that, all of these vectors fall in a 2 dimensional subspace in R^4 .

So, this is an important concept of subspace, which is very, very important for us from a data science viewpoint and I am going to explain to you why. We are interested in things like this, from a data science viewpoint. Now, the next question that we might ask is the following.

(Refer Slide Time: 23:30)

Finding basis vectors

2x4 < 8x4 2 basis 8(2x4)
19872 318 14x4

Evaluate the rank of the matrix

6	1	9	-3	3	14	11	7	2	7
5	2	4	1	-1	7	8	0	-3	7
8	3	7	1	-1	12	13	1	-4	11
11	4	10	1	-1	17	18	2	-5	15

Rank of the matrix is 2

Any two independent columns can be picked from the above matrix as basis vectors

Module 7 - Linear Algebra 16

So, this is the same as the previous slide, except that I have removed the dot dot dot. So, the way to think about this is let us say there is some data generation process, which is generating vectors like this, and the dot dot dots that I have left out, will also be generated in the same fashion, because those are also vectors that are being generated by the same data generation process.

So, I have certain data generation process and I am generating samples from that and I have done let us say 10 experiments. So, I have got these 10 samples and the other dots will be similar, now what I want to know is if you give me these, vectors in R⁴, how many basis vectors do I need to represent them? In the previous slide I had already shown you what the basis vectors are and then shown how I could generate many many linear combinations of just 2 in R⁴ to get a subspace. I am looking at an inverse problem here, where I do not know what are the vectors that are generating these samples, nonetheless I have got enough samples.

Let us say 10 and if I were to continue this experiment and if it was the same data generation process, I might get 20 samples 30 samples and so on; however, what I want to know is with these 10 samples, how do I find the basis vectors? So, we are going to use concepts that we have learned before to do this. If we were to stack all of these vectors in a matrix like this.

So, this is a first vector here, from here second vector and so on all the way up to the last vector and I say I have so many vectors, how many fundamental vectors do I need to represent all of these as linear combinations? It is a question that I am asking. The answer is straightforward this is something that we have already seen before, if you identify the rank of this matrix it will give you the number of linearly independent columns.

So, what that basically means is, if I get a certain rank for this matrix, then it tells me there are only so many linearly independent columns and every other column, can be written as a linear combination of those independent columns. So, while I have many many columns here, 1 2 all the way up to 10. The rank of the matrix will tell me, how many are fundamental to explaining all of these columns, and how many columns do I need.

So that I can generate the remaining columns as a linear combination of these columns, and as I have been mentioning again, if the data generation process remains the same as I add more and more columns to these, they will also be linear combinations of the columns that we identify here. So, when we go ahead and try to find the rank of this matrix, the rank of the matrix will turn out to be 2 and it will turn out to be 2 because, of the way we have generated this data.

Now, if you had generated these vectors in such a way that they are a linear combination of 3 vectors, then the rank of the matrix would have been 3. If you had generated these vectors in such a manner, that they are linear combinations of 4 linearly independent vectors, then the rank of the matrix would have been 4, but that would be the maximum rank of the matrix, because in R 4 you would not need more than 4 linearly independent vectors to represent all the vectors.

So, the maximum rank can be 4, the rank could be 1 2 or 3. If it is 1 then I have only 1 basis vector, if there are 2 there are 2 basis vectors 3 there are 3 basis vectors and so on. In this case since the rank of the matrix turns out to be 2, there are only 2 column vectors that I need to represent every column in this matrix. So, the basis set has size 2, is something that we have determined. The next question is the basis set is size 2, what are the actual vectors? What we can do is, we can pick any 2 linearly independent columns here and then those could be the basis vectors.

So, for example, I could choose this and this and say, this is the basis vector for all of these columns or I could choose this and this and this or this and this and so on. So, I can choose any 2 columns, as long as they are linearly independent of each other and this is something that we know, from what we have learned before, because we already know that the basis vectors need not be unique. So, I pick any 2 linearly independent columns that represents this data. Now, let me take a minute to explain why this is important from a data science viewpoint. I will just show you some numbers. Supposing, I have let us say 200 such samples and I want to store these 200 samples since each sample has 4 numbers, I would be storing 200 times 4 which is 8 numbers.

Now, let us assume we do the same exercise for these 200 samples and then we find that, we have only 2 basis vectors, which are going to be 2 vectors out of this set. What I could do is, I could store these 2 basis vectors that, would be 8 numbers which is 2 by 4 and for the remaining 198 samples, instead of storing all the samples and all the numbers in each of these samples, what I could do is for each sample I could just store 2 numbers right?

So, for example, if you take this sample, instead of storing all the 4 numbers, I could just store 2 numbers, which are the linear combinations that I am going to use to construct this. So, for example, since I have 2 basis vectors here, there is going to be some number α_1 times the basis vector, + α_2 times the second basis vector, which will give me this sample right?

So, instead of storing these 4 numbers, I could simply store these 2 constants and since I already have stored the basis vectors, whenever I want to reconstruct this, I can simply take the first constant and

multiply v 1 + the second constant multiply v 2 and I will get this number. So, I store 2 basis vectors which gives me 8 numbers and then for the remaining 198 samples, I simply store 2 constants. So, this would give me 396 + 8 404 numbers stored. I will be able to reconstruct the whole data set.

So, compare that with 800. So, I have half reduction in number. So, when you have vectors in multiple dimensions, let us say you have vectors in 10 dimensions 20 dimensions and the number of basis vectors, are much lower than those numbers. So, for example, if you have A₃0-dimensional vector and the basis vectors are just 3, then you can see the kind of reduction that you will get in terms of data storage. So, this is one viewpoint from data science. Why? It is very important to understand and characterize the data in terms of what fundamentally characterizes the data. So that you can store less, we can do smarter computations and there are many other reasons why we will want to do this, you can identify this basis to identify a model between this data, you can identify a basis to do noise reduction in the data and so on.

So, all of those viewpoints we will talk about as we go forward, with this data science course. In the next lecture, we will continue and then try and understand how we can use these concepts. The notion of basis vectors, the notion of orthogonality to understand concepts such as projections, hyper planes, half spaces and so on, which all are critical from a data science viewpoint. So, I will pick up from here in the next lecture

Thank you.

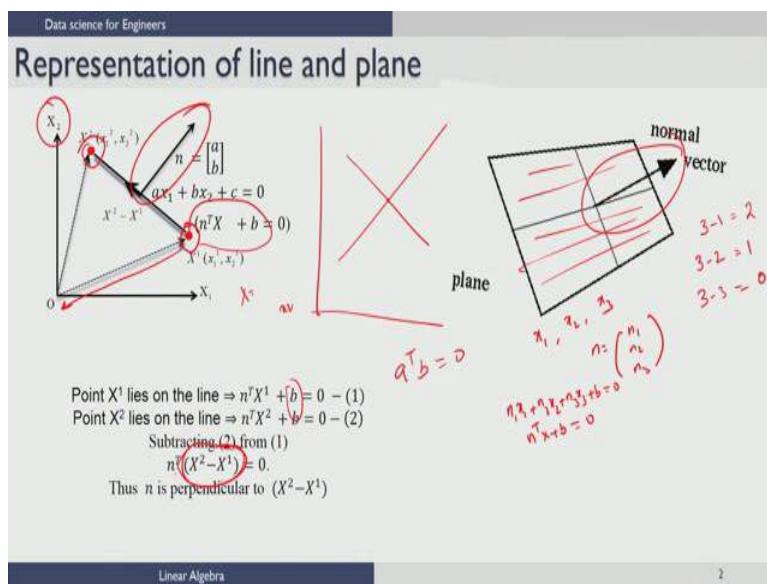
Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 16

**Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors
(Continued 1)**

So, let us continue with our lectures on linear algebra for data science. We will continue discussing distances hyperplanes, half spaces, eigenvalues and eigenvectors in this lecture, and the lecture that follows this lecture. So, what we are going to do is we are going to think about the equations in multi-dimensional space. And then think about what geometric objects that these equations represent.

(Refer Slide Time: 00:49)



So, let us look at what equations mean from a geometric viewpoint. To do this, let us start with 2 dimensions. Let us assume that we have a 2 dimensional space in X_1 and X_2 , and let us also assume that we have one equation that relates the variables; X_1 and X_2 which is $aX_1 + bx + c = 0$.

So, we want to understand what geometry this equation represents. It turns out in this case of the 2-dimensional space, this equation represents a line which is depicted here. So, a single equation in a 2-dimensional space represents a line. So, one might ask what does 2 equations represent.

And to understand this if you look at a picture like this, let us say I have one equation which is a line, let us draw the other equation which is also a line. Then if both of these equations have to be satisfied, then that has to be this intersecting point. So, 2 equations in 2 variables represent a point if these equations are solvable together.

Now, if you have no relationships between these variables, then we would say that we are representing all the points in the 2-dimensional space. And there is no relationship that constrains these points to either lie on a line or be a single point and so on. Now let us look at this equation, and then rewrite it in a form that is generally used. And that form is the following which is $n^T X + b = 0$, n is this column vector that is defined here. X is a vector of variables X_1 and X_2 . And if you do a one to one comparison between this equation and this equation, you will see that $b = c$. So, a general equation can be written in this form $n^T X + b = 0$.

Now, we want to understand what this n depicts in this example. Now if you look at this picture here, we have shown n as a normal to this line. And let us see why that is true. To see that, let us first start by looking at 2 points on the line. So, let us start with this point X_1 and then X_2 . Notice that both these points are on the line.

So, when I substitute X_1 into the equation for the line, it should satisfy it which is what is shown here $n^T X_1 + b = 0$ and when I substitute X_2 into the line equation that should also satisfy. So, $n^T X_2 + b = 0$. Now what you could do is you could subtract the first equation from the second equation. The b 's will get cancelled. You will have $n^T X_2 - X_1 = 0$. Now let us interpret this equation. From vector addition you know that if I have $X_2 - X_1$ is in this direction.

So, when I do $X_2 - X_1$ I am adding $X_2 1 - X_1$, which is equivalent to starting from here and going here ; which is basically through vector addition in the direction of this line. So, what this equation basically tells us is that this is in the direction of the line.

And from our orthogonality lecture we saw before, we said if $A^T b = 0$, then a and b are orthogonal. Since n^T this quantity is 0, and this quantity is in the direction of the line n has to be perpendicular to the line. So, this is a very important idea that we will use in data science quite a bit, when we looked at linearly separable classes. And then classifying classes that are linearly separable.

Now, if you want to extend this, and then ask the question, if I have one equation in a 3-dimensional space, what does that represent. Now the form of the equation will be very similar. You will have something like this here, and n now would become supposing you have 3 variables $X_1 X_2 X_3$. N could be $n_1 n_2 n_3$.

So, the same equation would be $n_1 X_1 + n_2 X_2 + n_3 X_3 + b = 0$. So, which is $n^T X + b = 0$. So, irrespective of what-ever is the dimension of your system, you can always represent a single linear equation in this form $n^T X + b = 0$.

Now, we ask the question as to what a single equation would represent in a 3 - dimensional space. And in a 3-dimensional space a single equation would represent a plane a 2-dimensional object. The way to see this is in 3 dimensions we have 3 degrees of freedom. If you write one equation you are taking away one degree of freedom. So, we are left with 2 degrees of freedom.

And a 2 degree of freedom object is basically a plane. And this is what is shown here. So, if I have one equation, that equation itself would represent this plane right here; which is what we see. And very similar to how we drew the normal to the line here, this n would represent a normal to the plane. So, this would represent a projection outside out of the plane orthogonal to the plane. So, that is what n would represent.

So, in 3 dimensions, one equation would represent a plane. And in 3 dimensions 2 equations would represent a line, because we have 3 degrees of freedom if you take away 2, then you have one, a one-dimensional object is a line. And if you have 3 equations, then you have $3 - 3 = 0$, the 0-dimensional object would be a point, and this would be a point as long as these 3 equations are consistent and solvable.

(Refer Slide Time: 08:10).

Projections

- We can define the projection (\hat{X}) of a vector (X) onto a lower dimension (two dimensions in the picture) mathematically as

$$\hat{X} = c_1 v_1 + c_2 v_2$$

- Using vector addition

$$X = c_1 v_1 + c_2 v_2 + n$$

v_1, v_2
 c_1, c_2
 $n^T v_1 = 0$
 $n^T v_2 = 0$
 $n^T n = 1$

Linear Algebra

Now, that we have talked about what equations represent and so on. One of the things that we are quite interested in and you will see this again and again in data science, as we teach some of the algorithms later such as principle component analysis and so on. We are always interested in projecting vectors onto surfaces. The reason why we are interested in doing this is, because many times we might want to represent data through a smaller set of objects or a smaller number of vectors. So, in some sense the data cannot be completely represented by these vectors.

So, we might ask the question as to what is the best approximation for this data point based on the vectors that I want to represent this data point with. So, this is a very important question that we will keep asking again and again. You will understand this in much more detail and clarity, once we talk about some of the data science concepts.

For now, I am just going to treat this mathematically, and then explain to you how we do projections and what are the equations that we can get for writing down projections. The interpretation for this and the use for this in data science is something that we will see as we go along this course later. So, let us take a very, very simple example. Let us assume that I have a plane which is shown here in this picture.

Since I have a plane basically we are looking at A₃-dimensional space. A plane has to be represented by 2 dimensions, because it is a 2-dimensional object. Let us assume that the basis vectors for this plane are v_1 and v_2 . We have already discussed what basis vectors are in a previous lecture. So, for a 2-dimensional object we will need 2 basis vectors.

So, let us assume these basis vectors are v_1 and v_2 . Just to recap what this basis vectors are useful for is that, any line on this plane, basically can be written as a linear combination of v_1 and v_2 . That is what we described before, that these basis vectors are enough to characterize every point or any vector on this 2-dimensional plane. So, any vector can be written as a linear combination of v_1 and a v_2 .

Now, the way this picture is drawn, you would see that this is the plane and I have let us say, a vector that is coming out of the plane. So, this is not clearly in the plane. So, it is projecting out. So, from the data science viewpoint if you want to make an analogy, what we are saying here is that, I have a data here X which is represented by this vector. I want to write this simply as only a function of v_1 and v_2 . So, in other words I want to represent this vector X, in a tool, I cannot do it exactly projecting out of the plane. So, I might ask what is the next best thing that I could do in this case. It turns out the next best thing to do would be to project this vector onto the plane, because ultimately, however I write this vector with only this 2 basis vectors it has to be on the plane.

Now, there are many vectors on the plane. I want to find what is the best projection for this onto this plane. So, a common sense idea would be to say, I want a point here which I write. And if this is the projection of this vector I want this distance to be minimized. So, you can see why that is. Think about this if you keep projecting it back to the plane, if this is the closest point if the vector is already in the plane, it would be the same vector that is also the product right. So, as soon as this vector goes up slightly outside the plane, I want it to be projected back. So, that it is closest to that point of projection. So, how do we explain these concepts mathematically? So, we do that here. First, \hat{X} is the projection of X onto lower dimension in this case 2 dimensions.

And since \hat{X} has to be in the lower dimension, We already know that it can be written as a linear combination of v_1 and v_2 . So, \hat{X} is $c_1 v_1 + c_2 v_2$. The c_1 and c_2 are yet to be determined. So, we do not know what those are. We are going to try and determine these 2 using this idea of projection. So, what we are going to say is that, if this is the projection, then the closest point from here would be when I draw a perpendicular or drop a perpendicular onto the plane. So, as long as these 2 points when I connect by vector n , that vector is perpendicular to this plane, then I would have found the closest point on this plane, which is what I am going for in terms of projections.

So, using vector addition, again we can start from here let us say, and this is x . So, X can be written as $\hat{X} + n$, which is what is written here. And \hat{X} has been expanded to be $c_1 v_1 + c_2 v_2$. Notice that while we write this, the fact that we are using a projection comes from this n being perpendicular to the plane. So, what does n being perpendicular to the plane mean? If n is perpendicular to the plane, then we know that $n^T v_1$ or $v_1^T n$ both are the same will be 0. Similarly, $n^T v_2 = v_2^T n$ will also be = 0. So, these are 2 facts that will know, if n is perpendicular to the plane. So, how are we going to use this to calculate c_1 and c_2 is what I am going to show you in the next slide.

(Refer Slide Time: 15:09)

• Projections onto general orthogonal directions (two dimensions in this case)

$v_1^T n = 0$

$$v_1^T (X - (c_1 v_1 + c_2 v_2)) = 0$$

$$v_1^T X - c_1 v_1^T v_1 = 0$$

$$\hat{X} = \frac{v_1^T X}{v_1^T v_1} v_1 + \frac{v_2^T X}{v_2^T v_2} v_2$$
 $c_1 = \frac{v_1^T X}{v_1^T v_1}$
 $c_2 = \frac{v_2^T X}{v_2^T v_2}$

Linear Algebra

4

So, let us first take this $v_1^T n = 0$; the first equation I wrote. Let me write n as this from the previous slide, because X was $c_1 v_1 + c_2 v_2 + n$, I simply move $c_1 v_1$ and $c_2 v_2$ to the other side. And I have this equation right here.

Now, when I expand this equation, I will get $v_1^T X - c_1 v_1^T v_1$ and I will also have another term which would be here $- c_2 v_1^T v_2$. Now as the first case I am going to show you how you do projections on 2 orthogonal directions. Now if these 2 directions are orthogonal the basis vectors themselves are orthogonal, then we know that this will be 0, that is the reason why this term drops out. And I have $v_1^T X - c_1 v_1^T v_1 = 0$. Take this to the other side, and then bring $v_1^T v_1$ to the denominator, then you will get $c_1 = v_1^T X$ divided by $v_1^T v_1$.

Now, you could use the same idea, and then do the calculations for $v_2^T n = 0$. And when you do this, again you use this fact that $v_2^T v_1$ or $v_1^T v_2 = 0$ because these are orthogonal directions. And then you will end up with this equation for c_2 , which will be $v_2^T X + v_2^T v_2$.

Once you get this, then you can back out the projection and the projection is c_1 times $v_1 + c_2$ times v_2 . So, this is how you project a vector on to 2 orthogonal directions, and this can be extended to 3 orthogonal directions 4 orthogonal directions and so on. Because all you will get if let us say it is 3 orthogonal directions then you will get $v_1^T X v_1^T v_1$ for c_1 , this is for c_2 and $v_3^T X$ divided by $v_3^T v_3$ for the third constant c_3 .

So, this is how you do projection. This is a very, very important idea, and this will be used in many many places in data science. So, it is worthwhile to clearly understand this.

(Refer Slide Time: 17:52)

Data science for Engineers

Projections: Example

$$X = [1 \ 2 \ 3]^T \checkmark$$

- Projecting this vector onto the space spanned by the vectors $v_1 = [1 \ -1 \ -2]^T$ and $v_2 = [2 \ 0 \ 1]^T$
- Thus, finding the projection onto the plane defined by v_1 and v_2 is

$$\hat{X} = \frac{[1 \ 2 \ 3] \begin{bmatrix} 1 \\ -1 \\ -2 \end{bmatrix}}{6} + \frac{[1 \ 2 \ 3] \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}}{5}$$

(Handwritten annotations: circled 'c1' above the first term, circled 'c2' above the second term, circled '6' under the denominator of the first term, circled '5' under the denominator of the second term, and a red bracket spanning both terms labeled '1 - 1 - 2 = 0').

Linear Algebra

5

Now, let us move on to doing an example for this projection. Let us take a very simple example, let us say I have vector 1 2 3 transpose; so, column vector. So, this is a vector in a 3-dimensional space.

Now, let us take 2 vectors and then make a plane. So, let us take vector v_1 which is 1 -1 -2, and v_2 which is 2 0 1. And then try and see whether I can project X on to these. Let us first find out whether these 2 vectors are orthogonal. So, to do that we have to do $v_1^T v_2$. So, I am going to do 1 -1 -2 2 0 1. So, this will be one times 2 -1 times 0 0 -2 times 1 2. So, 2 -2 is 0.

So, we know that these 2 vectors are orthogonal. So, we can use a formula that we had before. Now this formula is what we apply here. So, this is $v_1^T X$ transpose, sorry, this is $X^T v_1$ which is 1 -1 -2, and this should be $v_1^T v_1$. So, that will be one square + 1 square + 2 square. So, 1 + 1 + 4, 6. So, this is constant c_1 that we get, and this is multiplied by v_1 .

And if you look at the second term here. So, this is $X^T v_2$ which is 2 0 1. And this should be $v_2^T v_2$. So, this should be $2^2 + 0^2 + 1^2 = 5$. And we have this vector v_2 .

(Refer Slide Time: 19:50)

Data science for Engineers

Projections: Example

$$\hat{X} = \frac{-7}{6} \begin{bmatrix} 1 \\ -1 \\ -2 \end{bmatrix} + 1 \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$
$$\hat{X} = \begin{bmatrix} 5/6 \\ 7/6 \\ 20/6 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$


Linear Algebra

So, once you simplify this further you get the projection as the following. So, my original data vector 1 2 3, when it is projected onto a space spanned by these 2 basis vectors becomes this.

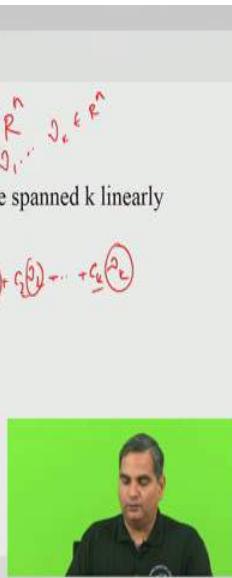
So, in other words, if I had a data point 1 2 3, and then I say I want to represent this with only 2 vectors that I had identified before whatever reason it might be, then the best representation is the following is what this projection says.

(Refer Slide Time: 20:26)

Data science for Engineers

Projection -Generalization

- Projections onto general directions
- Consider the problem of projection of X onto a space spanned by k linearly independent vectors

$$\hat{X} = \sum_{j=1}^k c_j v_j$$
$$\hat{X} = [v_1 \ \dots \ v_k] \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix}_{k \times 1}$$
$$\hat{X} = Vc$$


Linear Algebra

Now, we talked about projecting on to certain number of directions. And we also talked about projections when these directions are orthogonal. I am going to generalize this in the coming slides. So, that we have a result that is general and can be used in many places. So, I am going to look at how we can project this vectors onto general directions. So, let us consider the problem of projection of X onto space spanned by k linearly independent vectors,. Now I have dropped this notion of orthogonal here, I am simply saying these vectors are independent. As before since I want to project X on to k linearly independent vectors.

I am going to represent that projection as \hat{X} . And because this \hat{X} is in a space spanned by this k linearly independent vectors, I can write this as a linear combination of these k vectors; which is what I have written here. So, if you expand this you will get $c_1 v_1 + c_2 v_2$ and so on $+ c_k v_k$. Notice in this equation it is important to really understand this carefully. Notice in this equation $v_1 v_2 v_k$ are all vectors, and $c_1 c_2 c_k$ are scalar constants.

We can write this equation also in this form. Where, what we do is we stack these vectors, into a matrix. So for example, if X is in an n dimensional space R^n . Then we would assume that each of these vectors are also in R^n . So, v_1 to v_k are all element of R^n . And when you stack k vectors like this in a matrix, then you would get a matrix of dimension n by k .

And since there are k constants, which I have put in a vector. So, this would be a vector of dimension k by 1. And you can notice that this n by k times k by 1 will give you an n by 1 vector which is what this \hat{X} nonetheless, this n by one vector is in a space spanned by these k vectors linearly independent vectors.

Now this is an important thing to notice, if you go back and then say let me expand this, then basically you should get this. And this is another way of thinking about matrix multiplications, which is important to understand. So, let me illustrate this with some very, very simple examples so that we use this at later times.

(Refer Slide Time: 23:25)

Data science for Engineers

Projection -Generalization

- Projections onto general directions
- Consider the problem of projection of \hat{X} onto a space spanned by k linearly independent vectors

$$\hat{X} = \sum_{j=1}^k c_j v_j$$

$$\hat{X} = [v_1 \dots v_k] \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix}$$

$$\hat{X} = Vc$$

$$= c_1 v_1 + c_2 v_2 + \dots + c_k v_k$$

$X \in \mathbb{R}^n, v_i \in \mathbb{R}^n$

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

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

Linear Algebra

7

So, if I have let us say a matrix $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$, I will say, and I want to multiply this by $\begin{bmatrix} 0 & 0 \end{bmatrix}$. The standard way of doing this would be one times one + 1 times one and 0 times 1 + 1 times 1.

So, this will be 2 1, right. This is your standard matrix multiplication that you have seen. You can also interpret this slightly differently. You could say that this matrix multiplication is also one times this vector 1 0 + 1 times this vector. So, this is what we can think of this, matrix multiplication as. Now if

you notice this will also give you the result 2 times 2 0 0. So, what we are doing is there are many columns here and there are, these scalar constants much like this. So, when we multiply this this will be c_1 times the first column; much like, how we have written here. So, this will be c_1 times $v_1 + c_2$ times v_2 , all the way up to c_k times v_k .

So, this and this are same and you see that this is this. So, \hat{X} can be written as V times c ; where V is a matrix where all these basis vectors are stacked in columns. And c are the scalar constants which have been stacked as a single column. Now, let us proceed to identify the projection from here.

(Refer Slide Time: 25:00)

We then use the orthogonality idea. Remember, we have $X = \hat{X} + n$. That means, $n = X - \hat{X}$ and if \hat{X} has to be a projection, then n has to be a vector that is orthogonal to the space spanned by the k linearly independent vectors. For n to be orthogonal to a plane to a geometric object spanned by this k linearly independent vectors. N has to be orthogonal to every one of these vectors.

So, that is what we write here in a matrix form instead of writing $v_1 v_1^T X - \hat{X}$ is 0 $v_2 v_2^T X - \hat{X}$ is 0 and so on. So, we write this in a matrix form where we say v^T there I will have $v_1^T v_2$ transpose. All the way up to $v k^T$ times $X - \hat{X} = 0$.

Now, \hat{X} from the previous slide was v times c . So, $v^T X - v^T c = 0$. So, if I expand this I will get $v^T X - v^T v c = 0$. If I take this term to the other side, and then do the inverse. I will get $c = v^T v$ inverse $v^T X$. Whenever we take inverses we have to always make sure that we can actually identify an inverse. In this case I will be guaranteed to have an inverse for $v^T v$ if the columns of v are linearly independent. And the fact that we have chosen these basis vectors as linearly independent already, assures us that those are linearly independent.

So, this inverse is something that exists. Once we calculate this c we know \hat{X} is v times c . So, I simply plug this $v c$ back in and I get the expression for projection. So, this is how you do projection onto general directions. Now this is a very important idea that is used in several data science algorithms. In fact, this is a backbone for something called principal component analysis. And this is also used in many many other machine learning algorithms.

So, it is important to understand this idea very clearly. Now that we have understood projections, in the next lecture I will describe the notion of an hyper plane and half spaces. And then continue on to eigenvalues and eigenvectors. I will see you the next lecture.

Thank you.

Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture- 17

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors

We will continue with our lectures on linear algebra for data science. Today I will talk about hyper planes, half spaces and eigenvalues, eigenvectors and so on.

(Refer Slide Time: 00:27)

- Geometrically, hyperplane is a geometric entity whose dimension is one less than that of its ambient space.
- For instance, the hyperplanes for a 3D space are 2D planes and hyperplanes for a 2D space are 1D lines and so on.
- The hyperplane is usually described by an equation as follows
$$X^T n + b = 0$$

$$x_1 n_1 + x_2 n_2 + \dots + x_n n_n + b = 0$$
$$n_1 x_1 + n_2 x_2 + \dots + n_n x_n + b = 0$$
$$x^T n = 0$$

Let us start this lecture with hyper planes. Geometrically hyper plane is a geometric entity whose dimension is 1 less than that of its ambient space. So, what this means is, the following. For example, if you take the 3D space then hyper plane is a geometric entity which is 1 dimension less. So, its going to be 2 dimensions and a 2 dimensional entity in a 3D space would be a plane.

Now if you take 2 dimensions, then 1 dimension less would be a single dimensional geometric entity, which would be a line and so on. The hyper plane is usually described by an equation as follows, if I expand this out for n variables. So, I will get something like $X_1 n_1 + X_2 n_2 + X_3 n_3$ and so on $X_n n_n + b = 0$ in just two dimensions, you will see that this is $X_1 n_1 + X_2 n_2 + b = 0$ which is an equation of line. We have seen before the idea of subspaces. Hyperplanes in general are not

subspaces, however, if we have hyper planes of the form $X^T n = 0$; that is if the plane goes through the origin, then an hyper plane also becomes a subspace.

(Refer Slide Time: 02:03)

- We can observe that the equation can be evaluated for the two halfspaces
- It can be seen that

$$X^T n + b = 0 \quad \forall X \in \text{line}$$

$X^T n + b > 0 \quad \forall X \in \text{subspace in the } n \text{ direction } (X_3)$

$X^T n + b < 0 \quad \forall X \in \text{subspace in the } -n \text{ direction } (X_2)$

Now that we have described what a hyper plane is, let me move on to the concept of half space to explain the concept of half space. I am going to look at this 2 dimensional picture on the left hand side of the screen. So, here we have a 2 dimensional space in X_1 and X_2 and as we have discussed before an equation in two dimensions would be a line which would be a hyperplane. So, the equation to the line is written as $X^T n + b = 0$. So, for, in this two dimensions we could write this line as; for example, $X_1 n_1 + X_2 n_2 + b = 0$, while I have drawn this line only for part of this picture. In reality this line would extend all the way on both sides.

Now, you notice the following. You see when I extend this line all the way on both sides, then this whole two dimensional space is broken into two spaces, one on this side of a line and the other one on this side of a line. Now these two spaces are what are called the half spaces. Now the question that we have is the following.

If there are points on one half space and points on the other half space, is there some characteristic that separates them? For example, can I do some computations for all the points on one half space and get some value and some computation for all the points on the other half space and get some value and use that to make some decisions and that is a reason why we are interested in this half spaces from a data science viewpoint.

So, this question is of importance in a particular kind of problem called a

classification problem. Let me explain what that means. In fact, we are going to look at a very specific classification problem called binary classification problem. So, let us assume that I have, let us say in two dimensional space a data belonging to two classes.

For example, let us say I have data belonging to class one like this, and I call it class one and then I have data belonging to class two is something like this, call it class 2. So, these classes could be anything. So, for example, this could be a group of people who like South Indian restaurants and this could be a group of people, who do not like South Indian restaurants, and the coordinates X_1 and X_2 could be some way of characterizing people in terms of some attributes of these folks. Let us say we have taken a survey to say whether they like South Indian food or do not like South Indian food.

Now what we want to do is, if I give you the attributes of a new person, let us say that attribute falls here and then I ask you this question as to would this person like South Indian food or not like South Indian food and the answer would most likely be that this person will not like South Indian food, because this data point is very close to class 2.

Whereas if I gave you another point here for example, then you would come to the conclusion, this person is likely to like South Indian food. So, what we want to do is, we want to be able to evaluate cases like this. So, we want to somehow come up with a discriminating function between these two classes. So, one way to do that would be something like this; draw a line between these two classes and then say, if there is some characteristic that holds for this side of the line, which is what we called as a half space here. And if there is some characteristic that holds to this side of the line then we could use that characteristic as a discriminant function for doing this binary classification problem. So, that is the data science interest in understanding this topic in linear algebra.

Now let us proceed to see how we do this through some simple geometric concept. Let us go back to this picture and then ask the question as to how do I determine which side of a half plane or a half space, which half space does a point lie in. So, to understand this, what we are going to do is, we are going to take three points as shown here X_2 , X_1 and X_3 and ask the question as to how do I distinguish whether the point is on a line or to one half space or the other. So, the way we are going to do this, is the following. We are going to first look at this in little more detail and we know that when I write an equation of the form $X^T n + b = 0$, n is normal to this line, is something that we have already described.

However there is an important point to note. Here the normal could be defined in two ways, one is the normal is in this direction, the other

thing to do is to just take the opposite direction and then define a normal in this fashion also. So, it's important to know in which side normal is defined to understand this. For example, if I say this is a normal for an equation which is $X^T n + b = 0$.

(Refer Slide Time: 08:16)

Data science for Engineers

Halfspace

- We can observe that the equation can be evaluated for the two halfspaces
- It can be seen that

$$X^T n + b = 0 \quad \forall X \in \text{line}$$

$$X^T n + b > 0 \quad \forall X \in \text{subspace in the } n \text{ direction} (X_3)$$

$$X^T n + b < 0 \quad \forall X \in \text{subspace in the } -n \text{ direction} (X_2)$$

Linear Algebra

If I simply multiply this equation by -1, then I am defining a normal to the other side. So, this is an important point to remember. Now what we want to know is, where do these points X_1, X_2, X_3 lie to do this. What we are going to do is, we are going to evaluate a discriminant function or a function which is basically the equation of the line. So, what we want to do is.

(Refer Slide Time: 08:51).

Data science for Engineers

Halfspace

- We can observe that the equation can be evaluated for the two halfspaces
- It can be seen that

$$X^T n + b = 0 \quad \forall X \in \text{line}$$

$$X^T n + b > 0 \quad \forall X \in \text{subspace in the } n \text{ direction} (X_3)$$

$$X^T n + b < 0 \quad \forall X \in \text{subspace in the } -n \text{ direction} (X_2)$$

Linear Algebra

We want to understand what this will be, what this will be and what this will be. Now, when we look at point X_1 we know that the point lies on the line. So, this is going to be 0. So, this is straightforward. What we are interested in, is what happens to this quantity for X_3 and X_2 , and is there some way in which we can say that every point to one side of the line will have the same characteristic and every other point on the other side of the line will have a different characteristic. So, to do this, let us first look at $X_3^T n + b$ and then see what happens.

(Refer Slide Time: 09:44)

Data science for Engineers

Halfspace

- We can observe that the equation can be evaluated for the two halfspaces
- It can be seen that

$X^T n + b = 0 \forall X \in \text{line}$

$X^T n + b > 0 \forall X \in \text{subspace in the } n \text{ direction } (X_3)$

$X^T n + b < 0 \forall X \in \text{subspace in the } -n \text{ direction } (X_2)$

So, I want to know what this is. Notice in this picture I have defined a new point X on the line and then I have another vector which goes from X' to X_3 . Now X_3 is the vector that goes from here to here. From vector addition we know that I can write X_3 as X' , this + this $X' + Y'$. So, what I am going to do is, I am going to simply substitute this into the equation and then see what happens. So, I am going to have $X' + Y'^T n + b$.

This is what I want to evaluate. This will become $X'^T n + b + Y'^T n$. All I have done is, I moved b closer to this term to show you something. Now notice what happens to this term right here, since X' is on the line and the equation of line is $X^T n + b$ this has to go to 0. So, when we compute $X_3^T n + b$, we are simply left with this term right here. And if you notice this term you would see that this is a dot product between this vector and this vector, and the most important thing to note here is the following, as long as the point lies to this side, this side of the line then you would see whatever point you take, the angle between that point and the normal would be in the following ranges.

So, you take any point this side or this side. So, the angle between the normal and that point is going to be the following. So, supposing we look at this and then say; I am going to do this angle in this direction right. So, what you are going to notice is the following. If the point is between these two, then I am going to have a positive θ angle. Now the way you do this is the following.

So, you go like this. So, for this quadrant if you start with 0 here for this quadrant the angle is going to be between 0 and 90, and for this quadrant the angle is going to be between 270 and 360. So, if a point is this side, the angle between this vector and this normal is going to be between 0 and 90. And if the point is in this side the angle is going to be between 270 and 360. We also know that when I have dot products $A^T b$, I can also write this as magnitude of a magnitude of $b \cos \theta$, where θ is the angle between these two vectors.

So, we will look at all the points up to here. So, whatever is a point you have these angles and all of these angles are between 0 and 90. So, for any point between here and here in this whole space you are going to get a b , some angle between 0 and 90, and we know from our high school rule, all silver teacups $\cos \theta$ will always be positive. So, $A^T b$ is going to be positive; that means, this is going to be positive. Now when you get two points here then the angles are going to be between 270 and 360 which is in the fourth quadrant. Again using the same rule all silver teacups the fourth quadrant is $c \cos$. So, \cos is going to be positive. So, again you have $A^T b$ being positive.

So, irrespective of where the point is to this side of the line, when I take this $X_3^T n + b$, I am always going to get a positive value. Now by similar argument you can say for any point on the other side or the other half space, the angles are going to be between 90 to 180 here and 180 to 270 and as we know $\cos \theta$ for angles between 90 to 270 is negative.

So, any point on this side of the line or the half space the computation $X_2^T n + b$ is going to be less than 0. So, this is an important idea that that I would like you to understand. So, what this basically says is the following. If you were to simply take any point that I give you and then I evaluate $X^T n + b$, if that point is on the side of the normal half space then $X^T n + b$ will be positive, and if its on the half space in the opposite side then its going to be negative. And I already told you how this is important from a data science viewpoint.

(Refer Slide Time: 15:41)

Data science for Engineers

Hyperplanes and halfspaces: Example

- Let us consider a 2D geometry with $n = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$ and $b = 4$

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$X^T n + b = 0$$

$$[x_1 \ x_2] \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 4 = 0$$

$$x_1 + 3x_2 + 4 = 0$$

The hyperplane is the equation of a line

The halfspaces corresponding to this hyperplane are

$$x_1 + 3x_2 + 4 > 0 : \text{Positive halfspace}$$

$$x_1 + 3x_2 + 4 < 0 : \text{Negative halfspace}$$

Linear Algebra

4

So, let us consider simple 2D geometry and then let us take n as $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$ and b as 4. So, this would give me this equation for $X^T n + b$. Now let us say I take a point on three points for example. So, let me consider $(-1, -1)$ as one point, let us also consider $(1, -1)$ as another point and let us consider $(1, -2)$ as another point and then see what happens. So, when I take the point $(-1, -1)$ and I substitute into this $x_1 + 3x_2 + 4$. So, it will be $-1 - 3 + 4$. So, the point $(-1, -1)$ will lead to $-1 - 1$. Sorry will lead to 0. So; that means, the point $(-1, -1)$ is on the line, when I take the point $(1, -1)$. So, this is going to be $1 - 3 + 4$. So, this is going to be 2, so positive.

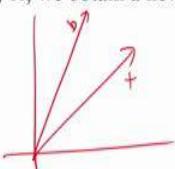
So, this is on in the positive half space and when I take the point $(1, -2)$ then I am going to get $1 - 6 + 4$ which is going to be -1 less than 0. So, this is in the negative half space. So, this is on the hyperplane of the line, this is on the positive half space and this is on the negative half space. So, that tells you how to look at different points and then decide which side of the hyperplane or which half space these points lie.

(Refer Slide Time: 17:35)

Data science for Engineers

Eigenvalues and eigenvectors

- We have previously seen linear equations of the form $Ax = b$
- What is the geometrical interpretation of this equation?
- We can make an interpretation as follows
 - When vector x is operated on by A , we obtain a new vector b with a different orientation



Linear Algebra

Now, that we have understood hyper planes and half spaces we are going to move on to the last linear algebra concept that I am going to teach in this module on linear algebra for data sciences. And once we are done with this topic, then we have enough information for us to teach you the various algorithms, commonly used algorithms or the first level algorithms in data science. So, let us look at this idea of eigenvalues and eigenvectors, we have previously seen linear equations of the form $Ax = b$. We have looked at it both algebraically and geometrically, we have spent quite a bit of time on looking at these equations algebraically. We talked about when these equations are solvable when there will be infinite number of solutions, how do we address all of those cases in a unified fashion and so on.

Now what we are going to do is, now that we know both vectors and so on, we are going to look at a slightly geometrical interpretation for this equation again and then explain the idea of eigenvalues and eigenvectors and then connect the notion of these eigenvalues and eigenvectors with the column space, null space and so on that we have seen before. So, this is very important, because these ideas are used quite a bit in data compression, noise removal, model building and so on. We will start saying I have this $Ax = b$ and A is an n by n matrix x is n by 1 and b is n by 1. So, this is the kind of system that we are looking at.

So, we are going only look at square matrices n by n . Now you can think of this as n equations and n variables. There is also another interpretation you can give for this which is the following, supposing I have a vector x something like this and if I operate A on this. So, by operating, I mean we define an operation as pre multiplying this vector by A . So, let us say I operate A on this vector which is Ax then I notice from this equation I get b , which is basically some other new direction

that I have. So, you can think about this as the following I can think about this as a equation, which tells me that when I operate A on x then I get a new vector b which is in a different direction from x. So, this is a very simple interpretation of this equation $Ax = b$, which is what is written here x, I send it through a and I define sending it through a as pre multiplying by A; so A times x equal b.

(Refer Slide Time: 20:45)

Data science for Engineers

Eigenvalues and eigenvectors

- Operator representation

The diagram shows a vector x (represented by a blue arrow) entering a box labeled A . The output is labeled $Ax = b$.

- The newly obtained b vector represents a new orientation. So we ask the following question
- Are there directions for a matrix A such that when the matrix operates on these directions they maintain their orientation save for multiplication by a scalar (positive or negative)?
- That is

The diagram shows a vector x (represented by a blue arrow) entering a box labeled A . The output is labeled $Ax = \lambda x$. To the right, two parallel red arrows represent the original vector x and the transformed vector λx , with a red bracket indicating they are parallel.

Linear Algebra

6

Now, that we have this interpretation, we ask the following question for a matrix A. Are there some directions which when you operate this A on they do not change their orientation. In other words I want to know if there are x vectors for matrix A such that when I operate A on x I get λx not b , λx here, the idea is because this is x, there is no change in orientation safe multiplication by a scalar. Now this multiplication scalar could be positive or negative, in which case we are talking about the following.

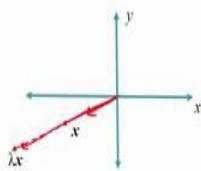
So, if this is x. So, when I operate A on x, since its in the same direction, its

either this way or this way and if λ is positive, it will be in this direction and if λ is negative, it will be in this direction and so on. So, the question is, would there be directions like this for all kinds of matrices is an interesting question, that you could ask for.

(Refer Slide Time: 22:04)

- The mathematical formulation of our question is

$$Ax = \lambda x$$
- The constant λ (*positive*) represents the amount of stretch or shrinkage the attributes x go through in the x direction
- The solutions (x) are known as eigenvectors and their corresponding λ are eigenvalues



Linear Algebra

Now let us focus on λ being positive, λ can be negative also. If λ is positive, then we see this equation and then notice that if λ is less than 1, then basically when I operate A on x the vector actually shrinks. So, if this is x , if λ is less than 1, this will be shrunk like this, and if λ is greater than 1 it will be at a higher magnitude than the original x vector.

Now the question is, for every matrix A , would there be A vectors like this x and what would be the scalar multiple and what is the use of all of this, is something that we should also address at some point as we go through this lecture. Now let me give you some definitions, this x are called eigenvectors and lambdas are called the eigenvalues corresponding to those eigenvectors. So, the questions that we are left with, are how do we find out that every matrix, whether it would have eigenvectors and how do I compute this eigenvectors and eigenvalues.

(Refer Slide Time: 23:26)

- We can find the eigenvalues as follows

$$Ax = \lambda x \quad A(n \times n); x(n \times 1)$$

$$Ax - \lambda Ix = 0 \quad \boxed{Ax - \lambda Ix = 0}$$

$$(A - \lambda I)x = 0 \quad \boxed{(A - \lambda I)x = 0}$$
- Thus the eigenvalues of the equation can be identified using

$$|A - \lambda I| = 0$$
- Substituting the eigenvalues in the original equation will help us find solutions for the eigenvector x

$x=0$ (trivial solution)
 x is in (the nullspace) ($A \neq I$)
 $(\text{rank}) + \text{nullity} = n$

So, to compute the eigenvalues we follow this procedure that I am going to outline now. So, the original equation is $A x = \lambda x$, what I could do is, I could bring this λx to this side, and then I get this equation $Ax - \lambda I x = 0$. So, this becomes $A - \lambda I$ times $x = 0$. Now, notice that this is basically A vector equation, because I have n by n vector this is n by 1. So, I have on the left hand side n by 1.

So, I have a vector here and I want 0s here. So, I want to find an x ; such that this is true. Now we have everything that we need to solve at this equation. So, what I am going to explain to you is the following. If I want to get an x which is not all 0, notice that if x is all 0, this is a solution right. So, $x = 0$ is a solution, but we are not interested in this solution, because this is what we call as a trivial solution.

We are not interested in this, we are only interested in solutions that we call as non trivial, at least one of the x s will have to be non 0. Now notice that if this equation is solvable, then x is in the null space of $A - \lambda I$ matrix. This is something that we have seen before, while we define the null space and we also know that the rank nullity theorem says the rank of the matrix + nullity = n which is the number of columns, we are looking at square matrices n by n matrices. Now we know that if there is even one vector x ; such that this is 0; that means, then rank of the null space is at least 1, and since the rank of the null space is at least one nullity is at least 1; that means, the rank of the matrix has to be less than n right, it cannot be n , if this is n nullity is 0, that means, there are no non trivial solutions.

So, if there needs to be a solution for x , then we know that the rank of the matrix $A - \lambda I$ has to be less than n ; that is the matrix $A - \lambda I$ is not a full rank matrix, and we know that if the matrix is not full rank then the determinant of that matrix has to be 0. So, in summary if we want a non trivial solution for x , then that necessarily means that this determinant $A - \lambda I$ has to be = 0.

Now once we solve for this equation and compute $A \lambda$, then we can go back and then substitute the value of λ here and then we have $A - \lambda I$ times $x = 0$, the way we have chosen λ is such that this matrix does not have full rank; that means, there is at least one vector in the null space, and using concepts that we have learned before we can identify this null space vector which would become the eigenvector.

(Refer Slide Time: 27:02)

Eigenvalues and eigenvectors: Examples

- Consider the following example with the given A matrix

$$A = \begin{bmatrix} 8 & 7 \\ 2 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 8 & 7 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \end{bmatrix}$$

$$|A - \lambda I| = \begin{vmatrix} 8 & 7 \\ 2 & 3 \end{vmatrix} - \lambda \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = \begin{vmatrix} 8 - \lambda & 7 \\ 2 & 3 - \lambda \end{vmatrix}$$

$$= 0$$

$$(8 - \lambda)(3 - \lambda) - 14 = 0$$

$$\lambda^2 - 11\lambda + 10 = 0$$

$$\lambda = (10, 1)$$

R Code

```
A = matrix(c(8,7,2,3), 2, 2, byrow=TRUE)
ev = eigen(A)
values = ev$values
```

Console output

```
> values
[1] 10 1
```



Linear Algebra

Let me illustrate this with an example here, let us consider the matrix A which is 8 7 2 3 and let us compute this determinant $A - \lambda I$. So, you get the following equation and you get a quadratic equation here. Notice an interesting thing here; if I have an n by n matrix, the determinant in λ would be an n th order polynomial. In this case I have A_2 by 2 matrix. So, the determinant is a λ function which is a quadratic, and if its 3 by 3 it will be cubic and so on. So, this opens out the possibility of a solution to this equation being complex also. So, this is an important point to note here though your original matrix A is real. The solution to your eigenvalue problem could be either real or complex, depending on the polynomial that you end up with. In this case we have chosen this example in such a manner that I get two real solutions and the real solutions are 10 and 1. So, I can easily see that this equation has solutions 10 and 1. So; that means, I have two

eigenvalues $\lambda_1 = 10$ and $\lambda_2 = 1$. Now how do I go ahead and calculate the eigen vectors corresponding to these eigen values.

(Refer Slide Time: 28:25)

Data science for Engineers

Eigenvalues and eigenvectors: Examples

$$\begin{bmatrix} 8 & 7 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- $\lambda = 1$

$$\begin{bmatrix} 8 & 7 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 8x_1 + 7x_2 \\ 2x_1 + 3x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$8x_1 + 7x_2 = x_1 \quad x_1 + x_2 = 0 \checkmark$$

$$2x_1 + 3x_2 = x_2 \quad x_1 + x_2 = 0$$

$$2x_1 + 2x_2 = 0 \rightarrow x_1 + x_2 = 0$$

$$x_1 + x_2 = 0$$

- Thus the eigenvector (unit) corresponding to $\lambda = 1$ is

$$X = \begin{bmatrix} 1 \\ \frac{\sqrt{2}}{2} \\ -1 \\ \frac{-\sqrt{2}}{2} \end{bmatrix}$$

So, let us illustrate this for $\lambda = 1$. So, I take this eigenvalue eigenvector equation now that I know $\lambda = 1$, this becomes $\begin{bmatrix} 8 & 7 \\ 2 & 3 \end{bmatrix} X_1 x$ is $X_1 + X_2$. Now this turns out into these two equations, and if you notice you take the first equation, the first equation is $8X_1 + 7X_2 = x_1$.

So, if I take X_1 to this side I get $7X_1 + 7X_2 = 0$, which is the same as $X_1 + x = 0$. If you take the second equation you will see that it is $2X_1 + 3X_2 = X_2$ which basically says $2x + 2X_2 = 0$, which also is $X_1 + X_2 = 0$. So, both these equations turn out to be the same. Now any solution where X_2 is the negative of X_1 would be a eigenvector, what we do is, the following of all of those solutions.

We also make sure that we get an eigenvector which has unit magnitude. So, if you notice here the eigenvector that we get, you notice that X_1 , and this is X_2 and you notice that X_2 is $-X_1$ or X_1 is $-X_2$, which is what will satisfy this equation and instead of picking any k as a solution here, we pick a k in such a way that the magnitude of this vector is 1. So, we know that the magnitude of this vector will be 1 by root 2 whole square + - 1 by root 2 whole square root which will be root of half + half which will be root of 1 = 1. So, that way we make this a specific eigenvector which is unit length. We could do the same thing for λ equal 10, by much the same procedure you will notice that you will get this equation here $7X_2$ is $2x$.

(Refer Slide Time: 30:30)

Data science for Engineers

Eigenvalues and eigenvectors: Examples

- $\lambda = 10$

$$\begin{bmatrix} 8 & 7 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10x_1 \\ 10x_2 \end{bmatrix}$$

$$\begin{bmatrix} 8x_1 + 7x_2 \\ 2x_1 + 3x_2 \end{bmatrix} = \begin{bmatrix} 10x_1 \\ 10x_2 \end{bmatrix}$$

$$7x_2 = 2x_1$$

- Thus the eigenvector (unit) corresponding to $\lambda = 10$

$$X = \begin{bmatrix} 7 \\ \sqrt{53} \\ 2 \\ \sqrt{53} \end{bmatrix}$$

R Code

```
A = matrix(c(8,7,2,3), 2, 2, byrow=TRUE)
ev = eigen(A)
vectors <- ev$vectors
> vectors
[1] [2]
[1,] 0.9615239 -0.7071068
[2,] 0.2747211 0.7071068
```

Linear Algebra

So, basically what you could do is, any vector which is such that if X_1 is kX_2 is $2k$ by 7 would satisfy this equation; however, what we do is, we choose this k in such a way that the magnitude of the eigenvector is 1. So, in this case 7 by $\sqrt{53}$ 2 by $\sqrt{53}$, if you do the magnitude of this you will see this is going to be $\sqrt{49 + 53 + 4 + 53}$, which will be $\sqrt{1} = 1$. So, you see that the magnitude is 1 and also this equation is basically satisfied by any eigenvector which is of this form k to k by 7 .

(Refer Slide Time: 31:33)

Data science for Engineers

Summary

- $Ax = b$ • Geometric interpretation
- $Ax = \lambda x$ • Eigenvalue-eigenvector equation
- λ • N eigenvalues from $|A - \lambda I| = 0$
- x • Eigenvectors, generally expressed in unit vector form

Linear Algebra

So, in summary for the eigenvalue eigenvector portion of this lecture, we started with $A x = b$ which has a geometric interpretation of A operating on x giving a new vector b . Now if we force this b to be λ

x some scalar multiple of x itself, where the scalar multiple could be either positive or negative, we get the eigenvalue eigenvector equation and to calculate the eigenvalue. What we do is, we calculate the determinant $A - \lambda I$. I set it to 0 for an n by n matrix, there will be an n th order polynomial that we need to solve which opens out to the possibility of the eigenvalues, being either real or complex. And once we identify the eigenvalues we can get eigenvectors as the null space of $A - \lambda I$ where λ is the corresponding eigenvalue.

In the next lecture what I will do is, I will connect this notion of eigenvalues and eigenvectors to things that we have already talked before in terms of column space and null space of matrices and so on. We already saw that the eigenvectors are actually in the null space of $A - \lambda I$, I am going to develop on this idea and then show you other connections between eigenvectors and these fundamental subspaces, and I will also allude to how this is a very important problem; that is used in a number of data science algorithms. So, I will see you in the next class.

Thank You.

Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 18

Linear Algebra - Distance, Hyperplanes and Halfspaces, Eigenvalues, Eigenvectors

This is the last lecture in the series of lectures on Linear Algebra for data science and as I mentioned in the last class, today, I am going to talk to you about the connections between eigenvectors and the fundamental subspaces that we have described earlier. We saw in the last lecture that the eigenvalue eigenvector equation results in

(Refer Slide Time: 00:40)

Data science for Engineers

Connections between eigenvectors, column space and null space

- We know that eigenvalues can be complex numbers even for real matrices
- When eigenvalues become complex, eigenvectors also become complex
- However, if the matrix is symmetric, then the eigenvalues are always real
- As a result, eigenvectors of symmetric matrices are also real
- Further, there will always be n linearly independent eigenvectors for symmetric matrices

$(A - \lambda I) = 0$
 $P_n(\lambda) = 0$
 $\lambda = 0$
 $A = A^T$
 $\lambda_1, \lambda_2, \dots, \lambda_n$
 $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \lambda = 1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

Linear Algebra

this equation having to be satisfied which is $A - \lambda I = 0$. In general, we also saw that, this would turn out to be a polynomial of degree n in λ , which basically means that even if this matrix A is real, because the solutions to a polynomial equation could be either real or complex, you could have eigenvalues that are complex.

So, for a general matrix, you could have eigenvalues which are either real or complex. And notice that since we write the equation $Ax = \lambda x$, whenever this eigenvalues become complex, then the eigenvectors are also complex vectors. So, this is true in general;

however, if the matrix is symmetric and symmetric matrices are of the form $A = A^T$, then there are certain nice properties for these matrices which are very useful for us in data science. We also encounter symmetric matrices, quite a bit in data science for example, the covariance matrix turns out to be a symmetric matrix and there are several other cases where we deal with symmetric matrices.

So, these properties of symmetric matrices are very useful for us when we look at algorithms in data science. Now, the first property of symmetric matrices that is very useful to us is; if the matrix is symmetric, then the eigenvalues are always real. So, irrespective of what that symmetric matrix is, this polynomial would always give real solutions for symmetric matrices. And as I mentioned before if this turns out to be real, then the eigenvectors are also real. Now, there is another aspect of an eigenvalues and eigenvectors that is important; if I have a matrix A and I have n different eigenvalues λ_1 to λ_n , all of them are distinct, then I will definitely have n linearly independent eigenvectors corresponding to them which could be v_1, v_2 all the way up to v_n ; however, if there are certain eigenvalues which are repeated. So, for example, if we take a case where eigenvalue λ_1 is repeated, then I could have some polynomial, which is like this.

So, the polynomial, original polynomial has eigenvalue; λ_1 repeated twice and then there is another $n - 2$ order polynomial, which will give you $n - 2$ other solutions. Now, in this case, when I have λ_1 repeated like this, then it could turn out that this eigenvalue either has 2 eigenvectors, which are independent or it might have just one eigenvector.

So, finding n linearly independent eigenvectors is not always guaranteed for

any general matrix and we already know that eigenvectors could be complex for any general matrix; however, when we talk about symmetric matrices, we can say for sure that the eigenvalues would be real, the eigenvectors would be real, further we are always guaranteed that we will have n linearly independent eigenvectors for symmetric matrices. It does not matter how many times the eigenvalues get repeated. One classic example of a symmetric matrix, where eigenvalues are repeated many times, so take identity matrix, something like this here, this identity matrix has eigenvalue $\lambda = 1$, which is repeated thrice.

But, it would have three independent eigenvectors; 1, 0, 0; 0, 1, 0 and 0, 0,

1. So, this is a case where eigenvalues repeated is thrice, but there are three independent eigenvectors. So, this is also an important result that we should keep in mind.

(Refer Slide Time: 05:04)

- Symmetric matrices have a very important role in data sciences
- In fact symmetric matrices of the form $A^T A$ or AA^T are often encountered
- Eigenvalues of matrices of the form $A^T A$ or AA^T while being real are also non-negative
- As discussed for general symmetric matrices, there will be n linearly independent eigenvectors for matrices of this form also
- What is the connection between the eigenvectors and the column space and null space of a (symmetric) matrix ?

$$\begin{aligned} (A^T A)^T &= A^T (A^T)^T \\ &= A^T A \end{aligned}$$



1.

Linear Algebra

And as I mentioned in the last slide, Symmetric matrices have a very important role in data sciences. In fact, symmetric matrices of the type, $A^T A$ or AA^T are often encountered in data sense computations. And notice that both of these matrices are symmetric. So, for example, if I take $A^T A^T$, this will be A^T ; A^T , which will be $A^T A$. So, the transpose of the matrix is the same. You can verify that,

AA transpose is also symmetric through the same idea. So, we know matrices of the form $A^T A$ or AA^T are both symmetric and they are often encountered; when we do computations in data science. And we know from the previous slide, I had mentioned that the eigenvalues of symmetric matrices are real, if the symmetric matrix also takes this form or this form.

We can also say that while the eigenvalues are real; they are also non-negative, that is they will be either 0 or positive, but none of the eigenvalues will be negative. So, this is another important idea that we will use; when we do data science, when we look at covariance matrices and so on. Also the fact that, this $A^T A$ and A^T are symmetric matrices; guarantees that there will be n linearly independent eigenvectors for matrices of this form also. So, what we are going to do right now is, because of the importance of symmetric matrices in data science computations, we are going to look at the connection between the eigenvectors and the column space a null space for a symmetric matrix. Some of these results translate to non-symmetric matrices also, but for symmetric matrices, all of these are results that we can use.

(Refer Slide Time: 07:04)

Data science for Engineers

Connections between eigenvectors, column space and null space

$\mathbf{Av} = \lambda v$

- What happens when the eigenvalues become zero?
 $\mathbf{Av} = \mathbf{0}$ $\lambda = 0$ is a multiplicity vector
- The eigenvectors corresponding to zero eigenvalues are in the null space of the matrix
- Conversely, if the eigenvalue corresponding to an eigenvector is not zero then that eigenvector cannot be in the null space
 $\lambda \neq 0$ and it must be a non-zero vector
if A is full rank

Linear Algebra

So, we go back to the eigenvalue eigenvector equation; Av is λv . And this result that we are going to talk about right now, is true whether the matrix is A symmetric or not. If $Av = \lambda v$, we ask the question, what happens when λ is 0? That is one of the eigenvalues becomes 0. So, when one of the eigenvalues becomes 0, then we have this equation which is $Av = 0$. So, we can interpret v as an eigenvector corresponding to eigenvalue 0.

We have also seen this equation before, when we talked about different sub-spaces for matrices; we saw that null space vectors are of the form $A\beta = 0$ from one of our initial lectures. You notice that, this and this form are the same. So, that basically means that, v which is an eigenvector corresponding like corresponding to eigenvalue, $\lambda = 0$, is a null space vector, because it is just of the form that we have here. So, we could say, the eigenvectors corresponding to 0 eigenvalues are in the null space of the original matrix A . Conversely, if the eigenvalue corresponding to an eigenvector is not 0, then that eigenvector cannot be in the null space of A . So, these are important results that we need to know.

So, this is how eigenvectors are connected to null space. If none of the eigen-values are zero, that basically means that the matrix A is full rank and; that means, that I can never solve $A v = 0$; and get non trivial v . So, it is not possible, if A is full rank. So, if A is full rank, I cannot solve for this and get non trivial v . So, whenever λ is; lambdas are such that, there are there is no eigenvalue that is zero; that means, A is full rank matrix; that means, there is no eigenvector such that $A v = 0$ which basically means that there are no vectors in the null space.

(Refer Slide Time: 09:40)

Data science for Engineers

Connections between eigenvectors, column space and null space

- Let us assume that there are r eigenvectors corresponding to zero eigenvalue
- This means that the null space dimension is r
- From rank-nullity theorem (discussed before), we know that the column rank should be $n - r$
- That is $n - r$ independent vectors are enough to represent all the vectors in the columns of the matrix (column space)
- What could be a basis for this column space or what could be the $n - r$ independent vectors?

*A_{nn} is Symmetric
r zero eigenvalues
n-r n-r zero eigenvalues
r eigenvectors
rank + nullity = n
rank = n-r*



Linear Algebra

Now, let us see the connection between eigenvectors and column space. In this case, I am going to show you the result; and this result is valid for symmetric matrices. Let us assume that I have a symmetric matrix A ; and the symmetric matrix A , we know will have n real eigenvalues. Let us assume that r of these eigenvalues are 0.

So, this r could be 0 also; that means, there is no eigenvalue which is zero. So, even then all of this discussion is valid. But as a general case, let us assume that r eigenvalues are 0. So, there are r zero eigenvalues. And since we are assuming this matrix is n by n , there will be n real eigenvalues of which r are 0. So, there will be $n - r$ non-zero eigenvalues. And from the previous slide, we know that the r eigenvectors corresponding to this r 0 eigenvalues are all in the null space ok. So, since I have r 0 eigenvalues, I will have r eigenvectors corresponding to this.

So, all of these r eigenvectors are in the null space which basically means that the dimension of the null space is r ; because there are r vectors in the null space; and from rank-nullity theorem, we know that rank + nullity = number of columns in this case n ; since there are r eigenvectors in the null space, nullity is r . So, the rank of the matrix has to be $= n - r$.

So, that is what we are saying here. And further we know that column rank = row rank; and since the rank of the matrix is $n - r$, the column rank also has to be $n - r$. This basically means that there are $n - r$ independent vectors in the columns of the matrix. So, one question that we might ask is the following; we could ask what could be a basis set for this column space? Or what could be the $n - r$ independent vectors that we can use as the columns subspace?

(Refer Slide Time: 12:24)

Connections between eigenvectors, column space and null space

- Notice that there are $n - r$ eigenvectors which are not in the null space
- We know that these are independent
- We also know that these vectors are a linear combination of all the column vectors – that is they are in the column space
- Further, we know that the dimension of the column space is $n - r$ (rank-nullity theorem)
- This implies that the eigenvectors corresponding to the non-zero eigenvalues form a basis for the column space

$A \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$

$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$

So, there are a few things that we can notice based on what we have discussed till now. First, notice that the $n - r$ eigenvectors that we talked about in the last slide, the ones that are not eigenvector is

corresponding to $\lambda = 0$; they cannot be in the null space; because λ is a number which is different from 0. So, these $n - r$ eigenvectors cannot be in the null space of the matrix A . So, let me write again. We are discussing all of this for symmetric matrices. We know, that all of this $n - r$ eigenvectors are also independent; because we said irrespective of what the symmetric matrix is, we will always get n linearly independent eigenvectors.

So, that means, these $n - r$ eigenvectors are also independent. We also know that each of these independent eigenvectors are going to be linear combinations of columns of A . To see this, let us look at this equation. So, let me write this out. So, I could call this as A , I am going to expand this v , into v_1, v_2, \dots, v_n , all the way up to v_n ; notice that these are components of v . We are just taking one eigenvector v and then these are the n components in that eigenvector. I can write this as λv and from the previous lecture of how to do this column multiplication and how to interpret this column multiplication, I said you could think of this as v_1 times the first column of $A + v_2$ times the second column of A ; all the way up to v_n types; n th column of $A = \lambda v$. Now in this equation, let me be very clear; these are scalars which are components in the eigenvector v ; these are column vectors; this is a first column of A , second column of A , this is n th column of A , this is again a scalar λ ; which is the eigenvalue corresponding to v .

So, this could be true for any of the $n - r$ eigenvectors; which are not in the null space of this matrix A . Now, take λ to the other side. So, you will have this equation as $v = v_1 by \lambda A_1 + \dots + v_n by \lambda A_n$. Again v_1 is a scalar λ is a scalar. So, these are all constants that we are using to multiply these columns. Now you will clearly see that, each of these eigenvectors; $n - r$ eigenvectors are linear combinations of the columns of A . So, there are $n - r$ linearly independent eigenvectors like this and each of this are combinations of columns of A . And we also know that the dimension of the column space is $n - r$. In other words, if you take all of these columns, A_1 to A_n ; these can be represented using just $n - r$ linearly independent vectors.

Now, when we put all of these facts together, which is the $n - r$ eigenvectors are linearly independent; they are combinations of columns of A ; and the number of independent columns of A can be only $n - r$. So, this implies that the eigenvectors corresponding to the non-zero eigenvalues for a symmetric matrix form a basis for the column space. So, this is the important result that I wanted to show you, with all of these ideas. Now again these results we will see and use as we look at some of the data science algorithms later.

(Refer Slide Time: 16:20)

Example

- Consider the following A matrix

$$A = \begin{pmatrix} 0.36 & 0.48 & 0 \\ 0.48 & 0.64 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

$$A^T = A$$

$$|A - \lambda I| = 0$$

$$P_A(\lambda) = 0$$

$$A = \lambda I$$

- Notice that this is a symmetric matrix

- The eigenvalues for this matrix are

$$\lambda = (0, 1, 2)$$

- The eigenvectors corresponding to these eigenvalues are

$$v_1 = \begin{pmatrix} -0.8 \\ 0.6 \\ 0 \end{pmatrix}, v_2 = \begin{pmatrix} 0.6 \\ 0.8 \\ 0 \end{pmatrix}, v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$A^T = A$

$$|A - \lambda I| = 0$$

$$P_A(\lambda) = 0$$

$$A = \lambda I$$



So, let us take a simple example to understand how all of these work. Let us consider a matrix which is of this form here; it is a 3 by 3 matrix. First thing that I want you to notice, that this is a symmetric matrix. So, if you do $A^T = A$. And we said symmetric matrices will always have real eigenvalues and when you do the eigenvalue computation for this, the way you do the eigenvalue computation is, you take determinant $A - \lambda I = 0$, then you are going to get a third order polynomial, you set it = 0; and then you calculate the three solutions to this polynomial and these would turn out to be the solution 0, 1, 2 and you take each of these solutions and then substitute it back in and then solve for $Ax = \lambda x$.

Then you would get the three eigenvectors corresponding to this which are given by this, this and this. I have noticed from our discussion before; since this is an eigenvector corresponding to $\lambda = 0$; so, this is going to be in the null space of this matrix A and these are the remaining 2; how do I get this 2? Which is $3 - 1$, n, n by n. So, it is A_3 by 3 matrix and nullity is 1; because there is only one eigenvector corresponding to $\lambda = 0$. So, I get 2 other linearly independent vectors. And in the last slide, when we were discussing the connections, we claim that these two eigenvectors will be in the column space or in other words, what we are claiming is that these three columns can simply be written as a linear combination of these two columns; and we are also sure that when we do A times v_1 , this will go to 0. So, let us verify all of this in the next slide.

(Refer Slide Time: 18:26)

Example

- From our prior understanding, the eigenvector corresponding to the zero eigenvalue will be in the null space
- We check that

$$A\mathbf{v}_1 = \begin{bmatrix} 0.36 & 0.48 & 0 \\ 0.48 & 0.64 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} -0.8 \\ 0.6 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Eigenvectors
to zero eigenvalue

- Interestingly, in the initial lectures, it was identified that the null space vector identifies a relationship between the variables
- Hence, the eigenvector corresponding to the zero eigenvalue can be used to identify the relationships among variables



So, let us first check A times \mathbf{v}_1 . So, this is a matrix, I have a times \mathbf{v}_1 here and you can quite easily see that when you do this computation, you will get this 0, 0, 0; which basically shows that this is the eigenvector corresponding to zero eigenvalue. Interestingly, in our initial lectures, we talked about null space and then we said the null space vector identifies a relationship between variables. Now, since this eigenvector is in the null space, the eigenvector corresponding to the eigenvector, corresponding to zero eigenvalue or eigenvectors corresponding to zero eigenvalues, identify the relationships between the variables because these eigenvectors are in the null space of the matrix.

So, it is an interesting connection that we can make. So, the eigenvectors corresponding to zero eigenvalue can be used to identify relationships among variables.

(Refer Slide Time: 19:33)

Example

- Let us now check if the other two eigenvectors shown below span the column space

$$\mathbf{v}_2 = \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix}, \mathbf{v}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

A_1, A_2, A_3
are linear
combinations
 \mathbf{v}_2 and \mathbf{v}_3

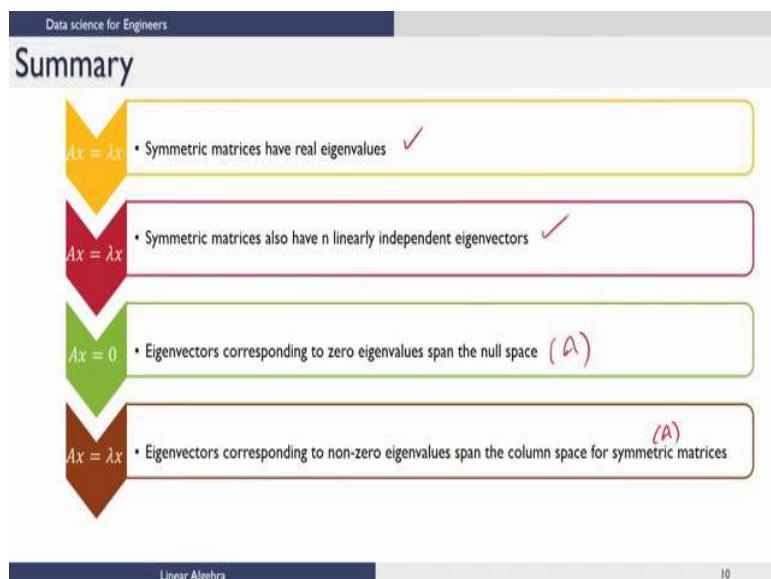
- This is demonstrated as below

$$\begin{aligned} A \begin{bmatrix} 0.36 \\ 0.48 \\ 0 \end{bmatrix} &= 6 * \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix} \\ A \begin{bmatrix} 0.48 \\ 0.64 \\ 0 \end{bmatrix} &= 8 * \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix} \\ A \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} &= 2 * \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned}$$



Now, let us do the last thing that we discuss. Let us now check the other two eigenvectors shown below. So, this is for the other two eigenvalues, span the column space. So, what I have done here is, I have taken each one of these columns from matrix A. So, this is column 1. So, this is A_1 , this is A_2 and this is A_3 . Column 1 is 6 times v_2 , column 2 is 8 times v_2 and column 3 is 2 times v_3 . So, we can say that A_1 , A_2 and A_3 are linear combinations of v_2 and v_3 .

(Refer Slide Time: 20:29)



So, v_2 and v_3 form a basis for this column space of matrix A. So, to summarize, we have $Ax = \lambda x$ and we largely focused on symmetric matrices in this lecture. So, we saw that, if we have symmetric matrices, they have real eigenvalues. We also saw that symmetric matrices have n linearly independent eigenvectors. We saw that the eigenvectors corresponding to zero eigenvalues span the null space of the matrix A and eigenvectors corresponding to nonzero eigenvalues span the column space of A for symmetric matrices that we described in this lecture. So, with this, we have described most of the important fundamental ideas from linear algebra that we will use quite a bit in the material that follows.

The linear algebra parts will be used in regression analysis, which you will see as part of this course. And many of these ideas are also useful in algorithms that do classification for example, we talked about half spaces and so on; and the notion of eigenvalues and eigenvectors are used pretty much in almost every data science algorithm, of particular note is one algorithm which is called the principle

component analysis which we will be discussing later in this course where these ideas of connections between null space, column space and so on are used quite heavily.

So, I hope that we have given you a reasonable understanding of some of the important concepts that you need to learn to understand some of the material that we are going to teach in this course and as I mentioned before, linear algebra is a vast topic. There are several ideas; how, how do these ideas translate, which ones of these are applicable or not applicable for non-symmetric matrices and so on. And from the previous lectures, how do we develop some of those concepts more can be found in many good linear algebra books; however, our aim here has been to really call out the most important concepts that we are going to use again and again in this first course on data science for engineers, more advanced topics in linear algebra will be covered when we teach the next course on machine learning where those concepts might be more useful in advanced machine learning techniques that we will teach. So, with this we close this series of lectures on Linear Algebra and the next set of lectures would be on the use of statistics in data science.

I thank you and I hope to see you back after you go through the module on statistics which will be taught by my colleague professor Shankar Narasimhan.

Data Science for Engineering
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 19
Statistical Modelling

This module on Statistical Modeling will introduce you the Basic Concepts in Probability and Statistics that are necessary for performing data analysis.

(Refer Slide Time: 00:26)

I	Random Variables, Probability Density Functions
1.1	Introduction
1.2	Random Phenomena
1.3	Probability Measure
1.4	Random variables, Probability density functions (pdfs)
1.5	Some common pdfs, Moments of a pdf
2	Estimation, Hypotheses Testing
2.1	Sample statistics
2.2	Estimation
2.3	Hypothesis testing

The module is divided into two parts: in the first part we will provide you an introduction to random variables and how they are characterized using probability measures and probability density functions, and in the second part of this module we will talk about how parameters of these density functions can be estimated and how you can do decision making from data using the method of hypothesis testing.

So, we will go on to characterizing random phenomena; what they are? And how probability can be used as a measure for describing such phenomena?

(Refer Slide Time: 01:03)

The slide is titled "Random Phenomena" and includes the GyanData logo at the top left. The main content is organized into two bullet points:

- Deterministic phenomenon: Phenomenon whose outcome can be predicted with a very high degree of confidence
 - Example: Age of a person (using date of birth stated in Aadhaar card)
- Stochastic phenomenon: Phenomenon which can have many possible outcomes for same experimental conditions. Outcome can be predicted with limited confidence
 - Example: Outcome of a coin toss

At the bottom of the slide, there is a footer bar with the text "Data Analytics" on the left and the number "4" on the right.

Phenomena can actually be either considered as deterministic phenomena whose outcome can be predicted with the high level of confidence can be considered as deterministic. For example, if you are given the information about the date of birth from an Aadhaar card of a person you can predict with a high degree of confidence the age of the person up to let us say number of days.

Of course, if you are asked to predict the age of the person to a hour or a minute the date of birth from an Aadhaar card is insufficient. Maybe you might need the information from the birth certificate, but if you want to predict the age with higher degree of precision, let us say to the last minute, you may not be able to do it with the same level of confidence. On the other hand, stochastic phenomena are those there are many possible outcomes for the same experimental conditions and the outcomes can be predicted with some limited confidence. For example, if you toss a coin you know that you might get a head or a tail but you cannot say with 90 or 95 percent confidence, it will be a head or a tail. You might be able to say it only with a 50 percent confidence if it is a fair coin.

Such phenomena we will call it as stochastic. Why are we dealing with stochastic phenomena.

(Refer Slide Time: 02:20)

GyanData Private Limited

Characterizing random phenomena

- Sources of error in observed outcomes
 - Lack of knowledge of generating process (model error)
 - Errors in sensors used for observing outcomes (measurement error)
- Types of random phenomena
 - Discrete: Outcomes are finite
 - Coin toss : {H, T}
 - Throw of a dice : {1, 2, 3, 4, 5, 6}
 - Continuous: Infinite number of outcomes
 - Body temperature measurement in deg F

Data Analytics



Because, all data that you actually obtain from experiments contain some errors these errors can either be, because we do not know all the rules that govern the data generating process, that is, we do not know all the laws we may not have knowledge of all the causes that affects the outcomes and therefore, this is called modeling error. The other kind of errors is due to the sensor itself. Even if we know everything the sensor that we use for observing these outcomes may themselves contain errors. Such errors are called measurement errors.

So, inevitably these two errors are modeled using probability density functions and therefore, the outcomes are also predicted with certain confidence intervals, which we derive. The types of random phenomena can either be discrete where the outcomes are finite - For example, in a coin toss experiment we have only two outcomes either a heads or tail or a throw of a dice where we have 6 outcomes - or it could be a continuous random phenomena which where we have an infinite number of outcomes such as the measurement of a body temperature which could vary between let us say 96 degrees to about 105 degrees depending on whether the person is running a temperature or not.

So, such continuous variable things which have random outcomes are called continuous.

(Refer Slide Time: 03:47)

The screenshot shows a presentation slide with the following content:

Sample space, events (discrete phenomena)

- Sample space
 - Set of all possible outcomes of a random phenomenon
 - Coin Toss : $S = \{H, T\}$
 - Two coin tosses: $S = \{HH, HT, TH, TT\}$
- Event
 - Subset of the sample space
 - Occurrence of a head in first toss of a two coin toss experiment $A = \{HH, HT\}$
 - Outcomes of a sample space are elementary events

At the bottom of the slide, there is a navigation bar with icons for back, forward, and search, followed by the text "Data Analytics" and the number "6".

Random phenomena we will try to describe all the notions of probability and so on using just the coin toss experiment. In this particular case we are looking at the discrete random variable or a random phenomena where we actually have a single coin toss whose outcomes are described by H and T. The sample space is the set of all possible outcomes. So, in this case the sample space consists of these two outcomes H and T denoted by the symbols H and T.

On the other hand if you are having two successive coin tosses, then there can be 4 possible outcomes either you might get a head in the first toss followed by a head in the second toss or a head in the first toss followed by a tail and so on. So, these are the four possible outcomes denoted by the symbol HH, HT, TH and TT and that constitutes what we call the sample space. The set of all possible outcomes an event is some subset of this sample space. For example, for the two coin toss experiment if we consider that and consider the event of receiving a head in the first toss then there are two possible outcomes that constitute this even space which is HH and HT we call this event a which is the observation of a head in the first toss.

Outcomes of the sample space for example, HH, HT, TH and TT can also be considered as events. These events are known as elementary events.

(Refer Slide Time: 05:14)

The slide is titled "Probability Measure". It lists the following points:

- Probability measure is a function that assigns a real value to every outcome of a random phenomena which satisfies following axioms
 - $0 \leq P(A) \leq 1$ (Probabilities are non-negative and less than 1 for any event A)
 - $P(S) = 1$ (one of the outcomes should occur)
 - For two mutually exclusive events A and B
 - $P(A \cup B) = P(A) + P(B)$
- Interpretation of probability as a frequency :
 - Conduct an experiment (coin toss) N times. If N_A is number of times outcome A occurs then $P(A) = N_A/N$

At the bottom right of the slide is a video frame showing a man with glasses and a light-colored shirt, sitting at a desk. The video player interface shows a play button, a progress bar, and a volume icon. The word "Data Analytics" is visible at the bottom left of the slide.

Now, associated with each of these events we define a probability. It is a measure which assigns a real value to every outcome of a random phenomena. When we assign this probability it has to follow certain rules. The first condition is that the probability we assign to any event should be bounded between 0 and 1 and that means, probabilities are non negative and it is less than 1. For any event that you might consider also the probability of the entire sample space should be = 1, which means 1 of the outcomes should occur; that is, what it means when you say $P(S) = 1$. And finally, the probability measure should also satisfy this condition that if you consider two exclusive events and say whether one or the other occurs.

The probability that either A or B occurs is the sum of $P(A)$ and $P(B)$; if A and B are exclusive events. The notion of exclusive events will be discussed in the subsequent slide. So, these are the though three rules that you should follow. When you assign a probability the easiest way of interpreting probability is as a frequency. For example, as an experimentalist you might want to do the coin toss experiment let us say 10,000 times N times and then count the number of times a particular outcome is observed. For example, let us say you are counting the number of times head occurs.

Let us say N_A is the number of times that the outcome corresponding to the head occurs, then the probability of head occurring can be defined as N_A by N. So, this you can see is bounded between 0 and 1, and if you look at the other outcome it will be $(N - N_A)$ by N and therefore, it will add up the probability of the sample space will be = 1. This way of defining how has a problem, because if you do

that toss 10,000 times instead of 1,000 times you might get a slightly different number.

So, the best way of interpreting this as a frequency is in the limit as N tends to ∞ and that is what we do as an assignment. If it is a fair coin then if we toss the coin a large number of times large meaning million billion times, then the probability of head occurring would be approximately = 0.5 and the probability of tail occurring will be approximately 0.5, if it is a fair coin and that is what we have assigned as probabilities.

(Refer Slide Time: 07:50)

The screenshot shows a presentation slide from GranData Private Limited. The title of the slide is "Exclusive and Independent Events". The content is organized into two main sections:

- Independent events**
 - Two events are independent if occurrence of one has no influence on occurrence of other
 - Formally A and B are independent events if and only if $P(A \cap B) = P(A) \times P(B)$
 - In a two coin toss experiment, the occurrence of head in second toss can be assumed to be independent of occurrence of head or tail in first toss, then $P(HH) = P(H \text{ in first toss}) \times P(H \text{ in second toss}) = 0.5 \times 0.5 = 0.25$
 - Mutually exclusive events**
 - Two events are mutually exclusive if occurrence of one implies other event does not occur
 - In a two coin toss experiment, events {HH} and {HT} are mutually exclusive $\Rightarrow P(HH \text{ and } HT) = P(HH) + P(HT) = 0.25 + 0.25 = 0.5$

A small video player window in the bottom right corner shows a man speaking. The slide footer includes navigation icons and the text "Data Analytics".

Now, we can go on to define two important types of events what is called the independent set of events. Two events are said to be independent, if the occurrence of one has no influence on the occurrence of other. That is, even if first event occurs we will not be able to make any improvement about the predictability of B if A and B are independent formally. In probability it is the way we consider two events to be independent is if $P(A \cap B)$ which means A joint occurrence of A and B can be obtained by multiplying their respective probabilities which is $P(A)$ into $P(B)$.

Let us illustrate this by A by A example of the coin toss experiment. Suppose you toss the coin twice. Now if you tell me that the first toss is a head then does it allow you to improve the prediction of a head or a tail in the second toss? Clearly you will say well does not matter whether the first toss was a head or a tail the probability of head occurring as the second toss is still 0.5. That means, information you provide me about the first toss has not changed my predictability of head or tail in the second toss.

So, if we look at the joint probability of two successive heads which is the head in the first toss and the head in the second toss, because we consider them as independent events we can obtain the probability of this two successive heads as a probability in the first toss of head in the first toss multiplied by the probability of head in the second toss which is 0.5 into 0.5 and 0.25.

So, all the four outcomes in the case of two coin toss experiment, we will have a probability of 0.25, whether you get two successive heads or two successive tails or a head or a tail or a tail in the head all will be 0.25 and this way we actually assign the probabilities for the two coin toss experiment from the probability assignment of a single coin toss experiment. Now, mutually exclusive events are events that preclude each other. Which means, if you say that event A has occurred then it implies B has not occurred, then A and B are called mutually exclusive events one excludes the other occurrence of one excludes the other.

So, let us look at the coin toss experiment again. Two coin tosses in succession we can look at the event of two successive heads as precluding the occurrence of a head followed by a tail. If you tell me two successive heads have occurred, it is clear that the event of head followed by a tail has not occurred. So, these are mutually exclusive events. The probability of either receiving two successive heads or a head and followed by a tail can be obtained in this case by simply adding their respective probabilities because they are mutually exclusive events. So, we can say the probability of either a HH or a HT which is nothing but the event of a head in the first toss is simply 0.25 + 0.25 which is 0.5 which is obtained by a basic laws of probability of mutually exclusive events.

(Refer Slide Time: 11:11)

GyanData Private Limited

Some rules of probability

- Following important probability rules can be proved using Venn diagrams

$S = \square$ $A = \text{Red circle}$ $B = \text{Blue circle}$
All outcomes are equally likely

If A^c is the complement of event A^c ,
 $P(A^c) = P(S) - P(A) = 1 - P(A) = 0.5$

If $B \subseteq A$, $P(B) \leq P(A)$; $0.25 < 0.5$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \\ = 0.5 + 0.5 - 0.5 \cdot 0.5 = 0.75$$



Data Analytics

9

Now, there are other rules of probability that we can derive and these can be done using Venn diagrams. So, here we have illustrated this idea of using Venn diagram to derive probability rules by for the 2 coin toss experiment. In the two coin toss experiment the sample space consists of 4 outcomes denoted by HH, HT, TH and TT.

We are interested in the event A, which is a head in the first toss. This consists of two outcomes HH and HT, which is indicated by this red circle. A compliment is the set of all events that exclude A which is nothing but the set of outcomes TH and TT is known as a complement. Now from the rules of probability you can actually derive the probability of a compliment is nothing but the probability of the entire sample space – $P(A^c)$, which is one. Which is because probability of S is $1 - P(A)$, notice the $P(A)$. In this case is the $P(HH)$ which is $0.25 + P(HT)$ which is $0.25 = 0.5$. So, eventually we get the probability of a compliment with this TH and TT = 0.5. This could have also been computed by looking at the $P(TH) + P(TT)$ which is 0.5.

So, it verifies that $P(A^c) = 1 - P(A)$. Now you can consider a subset, in this case even be denoted by the blue circle of two successive heads notice two successive heads it is a subset of receiving a head in the first toss which is A event A. So, we can claim that if B is a subset of A, then the $P(B)$ should be less than the $P(A)$. You can verify that the $P(B)$ is two successive heads which is 0.25 is less than the $P(A)$ which is 0.5. You can also compute the probability joint probability of two events A and B, which is not joint probability, but the $P(A) \text{ or } B$ which is given by $P(A \cup B)$ can be derived as $P(A) + P(B) -$ the probability of joint occurrence of A and B. Let us consider this example of receiving

a head in the first toss which is event A and receiving a head in the second toss which is event B.

So, receiving a head in the second toss consists of two outcomes HH and TH denoted by the blue circle. Now notice that they have a common event of two successive heads which belongs to both A and B. So, A and B are not mutually exclusive, but have a common outcome. Now in order to compute the $P(A)$ or B which means either I receive a head in the first toss or I receive a head in the second toss, then this comes to three outcomes and together gives you a probability of 0.75 which we can count from the respective probabilities of HT, HH and TH, but this can also be derived by looking at the $P(A)$ which is $0.5 + P(B)$; which is $0.5 - P(A \cap B)$ which is the probability of HH which itself can be computed by multiplying the probability of receiving a first hedge in the first toss and the probability of head in the second toss which is 0.25.

So, overall gives you $1 - 0.25$ which is 0.25 which is what we can derive by counting the respective adding up the respective probabilities of the mutually exclusive events HT, HH and TH. So, such rules or things can be proved by using Venn Diagrams in a simple manner.

(Refer Slide Time: 15:01)

The slide is titled "Conditional Probability" and is part of a presentation by GranData Private Limited. It contains the following content:

- If two events A and B are not independent, then information available about the outcome of event A can influence the predictability of event B
- Conditional probability
 - $P(B | A) = P(A \cap B)/P(A)$ if $P(A) > 0$
 - $P(A | B)P(B) = P(B | A)P(A)$ - Bayes formula
 - $P(A) = P(A | B)P(B) + P(A | B^c)P(B^c)$
- Example: two (fair) coin toss experiment
 - Event A : First toss is head = {HT, HH}
 - Event B : Two successive heads = {HH}
 - $P(B) = 0.25$ (no information)
 - Given event A has occurred $P(B|A) = 0.5 = 0.25/0.5 = P(A \cap B)/P(A)$

Now that is an important notion of conditional probability, which is used when two events are not independent. So, if two events are not independent; then if you can provide me some information about A, it will influence the predictability of B vice versa if you tell me some information about the occurrence of B then this information will improve the predictability of A, if the two events are not independent.

So, we define what is called the conditional probability, that is, the probability of event B occurring given that event A has occurred can be obtained by this formula which is the $P(A)$ and B simultaneously occurring divided by the $P(A)$ occurring.

Given course, assuming that $P(A)$ is greater than 0. Now using this notion of conditional $P(B)$, given A and this formula we can derive what is called the Bayes rule, which simply says the conditional $P(A|B)$ multiplied by the $P(B)$ is the conditional $P(B|A)$ multiplied by $P(A)$. This rule can be easily derived from the first rule by simply interchanging A and B and deriving the conditional $P(A|B)$ multiplied by $P(B)$, which is the $P(A \cap B)$ and right hand side. In this also is $P(A \cap B)$, both of these are equal to A intersect $P(A)$ intersection B . We can also derive another rule for $P(A)$ which is $P(A|B) P(B) + P(A|B)^c P(B)^c$

Notice that B and B complement are mutually exclusive and therefore conditional event to A given B and A given B complement are mutually exclusive and therefore, you are able to add the probabilities. So, let us illustrate this by a two coin toss experiment. Let us consider the event A which is a head in the first toss and event B which is two successive heads. Notice that A and B are not independent and which you can easily verify by computing the probabilities also. So, if you do not give me any information about event A , I will tell you that the probability of receiving two successive heads is 0.25, which is the probability of heads in the first toss multiplied by the probability of head in the second toss.

However, if you tell me that you are observed event A ; that means, that the first toss is a head, in this case then the probability of event B is actually improved. I can tell now there is a 50 percent chance of getting probability event B , because you have already told me that the first toss is a head. So, notice that I can compute this probability conditional $P(B)$ given A . Using the first rule which is $P(A \cap B)$ which is 0.25 divided by the $P(A)$ which is 0.5. So, this $P(B)$ given A is 0.5 which has improved my ability to predict B , because I have used some information you have given about point event A . Now, if B and A were totally independent, then information that you are provided to A will not affect the probability of predicting predictability of B it would have remain the same in this case it does not remain the same.

(Refer Slide Time: 18:53)

The slide has a header 'GyanData Private Limited' and a title 'Example'. The main content is enclosed in a rounded rectangle:

In a manufacturing process 1000 parts are produced of which 50 are defective. We randomly take a part from the day's production

- Outcomes : {A=Defective part B = Non-defective part}
- $P(A) = 50/1000, P(B) = 950/1000$
- Suppose we draw a second part without replacing the first part
 - Outcomes : {C = Defective part D = Non-defective part}
 - $P(C) = 50/1000$ (no information about outcome of first draw)
 - $P(C | A) = 49/999$ (given information that first draw is defective)
 - $P(C | B) = 50/999$ (given information that first draw is non-defective)
 - $P(C) = 49/999 * 50/1000 + 50/999 * 950/1000 = 50/1000$
 - $P(A | C) = P(A \cap C)/P(C) = P(C | A)P(A)/P(C) = 49/999$

So, B and A are not independent. We will illustrate again a example all these ideas of probability.

Suppose we have a manufacturing process where we actually have manufactured 1,000 parts out of which 50 parts are defective. Now from the collection of parts produced in a day, we randomly choose one part and ask this question would this part that we have selected picked, would it be a defective part or what is the probability it will be a non defective part.

Clearly, because there are 50 defective parts and each of these parts can be uniformly picked, we know that the $P(A)$ is the number of defective parts divided by the total number of parts which is 50 by 100, 1,000. On the other hand, the probability of picking a non defective part is the complement of this, which is 950 divided by 1,000. Now let us assume that we have picked one part kept it aside and we draw a second part without replacing the first part into the pool. We are interested in the outcome whether the second part that we have picked is it a defective part or a non defective part.

Suppose you do not tell me anything about what happened in the first pick, then I will say that the probability of picking as defective part even in the second is unchanged it is 50 by 1,000. Let us see how this comes about. At this point it may not be clear that it is 50 by 1,000, but we will show this formally. Now let us assume I give you some information about A. Suppose, I tell you that the first part that you do was a defective part, then clearly the total number of defective parts have decreased to 49 and the total number of parts has decreased to 999.

So, the probability of picking a defective part in the second pick given that you picked a defective part in the first pick is 49 by 999. On the other hand, if you tell me that the first draw is non defective which means the total number of parts again as reduce to 999, but the number of defective parts in the pool still remains at 50.

So, the probability of picking as defective part in the second round given that the first pick was non defective is 50 by 999. Now, according to the rules of conditional probability we can compute the $P(C)$, by $P(C | A)$; which is 49 by 999 multiplied by the $P(A)$; which is 50 by 1,000 + the $P(C | A)^c$. Remember, A complement is nothing, but B.

So, the $P(C | A)^c$ is 50 by 999 multiplied by the $P(A)^c$ which is nothing, but 950 by 1,000, which we have actually shown in the first case. So, if you add up all these probabilities. You will find that you get 50 by 1,000 which is that if you do not give me any information about. What has happened in the first pick? Whether you replace the part or whether you do not replace the part the probability of picking a defective part in the second ring is 50 by 1,000.

Non obvious, but it is the same if you do not give me any information about the first pick it does not matter, whether you replace the part or you do not replace the part your predictability your ability to predict still remains the same 50 by 1,000. On the other hand clearly, if you give me some information I am able to change the probably either decreases or increases depending on what was the outcome of the first pick.

Now it is very interesting to actually ask the inverse question. If you tell me some information about the second pick would it actually change your ability to predict the outcome of the first pick? It turns out it does, because these are not independent events. You can ask the question, what is the probability of getting a defective part in the first pick given that you had a defective pick in the second round.

Now, if you apply again the rules of conditional probability. You can say $P(A|C)$ is $P(A \cap C)$ divided by $P(C)$, but $P(A \cap C)$ can be written as probability of C given a conditional probability C given multiplied by $P(A)$. So, the whole thing is $P(C | A)$ multiplied by $P(A)$ divided by $P(C | A)$. We have computed as 49 by 999; $P(A)$ is 50 by 1,000 divided by probability of C which is 50 by 1,000.

So, finally, I get $P(A)$ given C is 49 by 99. Notice $P(A)$ itself is 50 by 1,000, but it has now reduced to 49 by 999, because you told me that the second pick was a defective part clearly it seems to be that somehow the first pick information is dependent on the second pick information which is obviously true because you have not done a replacement here. If you have done A replacement on the other hand, you will find that the outcome of C will be completely independent of

outcome of A and you will not be able to improve or decrease the predictability of A in the first pick.

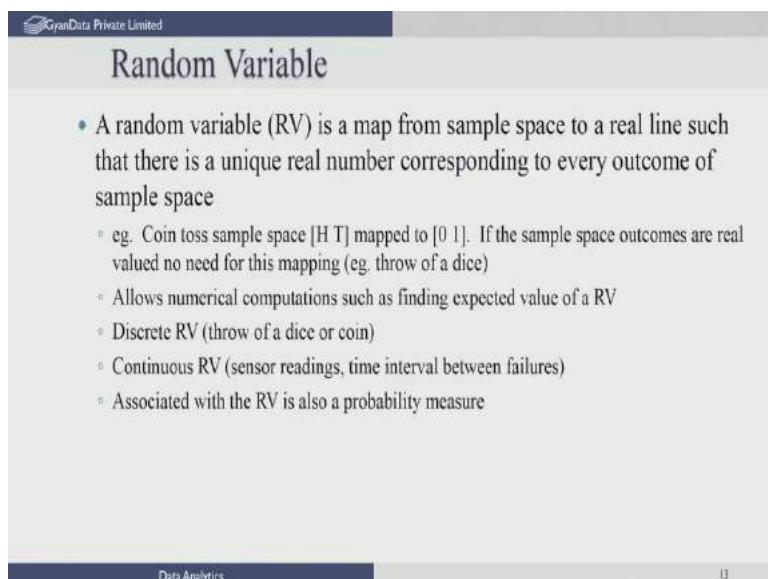
So, all these ideas of conditional probability independent events; mutually exclusive events will be repeatedly used in the application of data analysis and we will see how.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 20
Random Variables and Probability Mass/Density Functions

In the previous lecture I introduced the concept of Random Phenomena and how such phenomena can be described using probability measures. In this lecture, I am going to introduce the notion of random variables and the idea of probability mass and density functions. We will also see how to characterize these functions and how to work with them?

(Refer Slide Time: 00:37)



GranData Private Limited

Random Variable

- A random variable (RV) is a map from sample space to a real line such that there is a unique real number corresponding to every outcome of sample space
 - eg. Coin toss sample space [H T] mapped to [0 1]. If the sample space outcomes are real valued no need for this mapping (eg. throw of a dice)
 - Allows numerical computations such as finding expected value of a RV
 - Discrete RV (throw of a dice or coin)
 - Continuous RV (sensor readings, time interval between failures)
 - Associated with the RV is also a probability measure

So, a random variable is a function which maps the outcomes of a sample space to a real line. So, there is a unique real number that is associated to every outcome in the sample space. Why do we need a notion of a random variable?

Let us take the example of a coin toss experiment in which the outcomes are denoted by symbols H and T. H refers to the head and T refers to the tail. Unfortunately, we will not be able to do numerical computations with such representation therefore, what we do is to map these outcomes to points on the real line. For example, we map H to 0 and tail to 1. The random variable or function that maps the outcomes

of a sample space to this real life that is what we are referring to as a random variable.

Now, if the outcomes of random phenomena are already numbers such as the true of a dice, then we do not need to do this extra effort. We can work with the outcomes themselves in that case and call them random variables. We will see how we can do numerical computations once we have such a map and before in a way similar way we have discrete and continuous random phenomena. We will describe discrete random variables that maps functions of discrete phenomena to the real line and continuous random variable which maps outcomes of a continuous random phenomena to intervals in the real life.

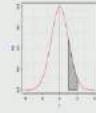
So, just as we had a probability measure to characterize outcomes of random phenomena, we will have a similar probability measure that characterizes or is associated with this random variable.

(Refer Slide Time: 02:34)

 GyanData Private Limited

Probability Mass/Density Functions

- For a discrete RV the probability mass function assigns a probability to every outcome in sample space
 - Sample space of RV (x) for a coin toss experiment: $[0, 1]$.
 - $P(x=0) = 0.5; P(x=1) = 0.5$
- For a continuous RV the probability density function $f(x)$ can be used to assign a probability to every interval on a real line
 - Continuous RV (x) can take any value $[-\infty, \infty]$
 - $\int_a^b f(x) dx$ (Area under the curve)
 - Cumulative density function $F(x) = \int_{-\infty}^x f(x) dx$



$$F(b) = P(-\infty < x < b) = \int_{-\infty}^b f(x) dx$$

Slide number: 14

The notion of a probability mass or a density function is a measure that maps the outcomes of a random variable to values between 0 and 1. For example, if we take the coin toss experiment and associate a random variable x whose outcomes are 0 and 1, then we associate a probability for $x = 0.5$ and a probability for $x = 1$ which is another 0.5. Notice that this association should follow the same laws of probability that we described in the last lecture. This is a fair coin and that is why the outcomes are given equal probability.

Now, in the case of a continuous random variable, we define what is known as a probability density function, which can be used to compute the probability for every outcome of the random variable within an

interval. Notice, in the case of a continuous random variable, there are ∞ of outcomes and therefore, we cannot associate a probability with every outcome. However, we can associate a probability that the random variable lies within some finite interval. So, let us call this random variable x which can take any value in the real line from $-\infty$ to ∞ , then we define the density function $f(x)$, such that the probability that the variable lies in an interval a to b is defined as the integral of this function from a to b .

So, the integral is an area. So, the area represents the probability. So, this is denoted on the right hand side, you can see a function and here we show how the random variable the probability that the random variable lies between -1 to 2 is denoted by the shaded area. That is how we define the probability and $f(x)$ which defines this function is called the probability density function.

Again, since this has to obey the laws of probability the integral from $-\infty$ to ∞ of this function or the area under the entire curve should be $= 1$ and obviously, the area is non zero therefore it obeys the second law also we actually described in the last lecture. We can also define what is called the cumulative density function, which is denoted by capital F and this is the probability that the random variable x lies in the interval $-\infty$ to b for every value of b you can compute this value function value and this is nothing, but the integral between $-\infty$ and b of this density function $f(x) dx$.

That is known as the cumulative density function for $b = -\infty$ the cumulative density function value will be 0 and $b = \infty$ you can verify that the cumulative density function takes the value one. So, this cumulative density function goes from 0 to 1 as the upper limit of the interval goes from $-\infty$ to $+\infty$.

(Refer Slide Time: 05:55)

Binomial Mass Function

- Probability of obtaining k heads in n coin tosses with p the probability of obtaining a head in any toss
- RV x represents number of heads obtained
 - Sample space : $[0, 1, \dots n]$
 - Formula: $f(x = k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$
 - One outcome: $\frac{HHT\ldots HTT\ldots T}{H\text{ Head } T\text{ Tail } H\text{ Head } T\text{ Tail } \dots}$
 - PMF characterized by one parameter p
 - For large n it tends to a Gaussian distribution

Binomial mass function for $n = 20, p = 0.5$

Now, let us look at some sample probability mass functions. Let us take the case of n coin tosses of an experiment where we have do n coin tosses and we are asking the question what is the probability of obtaining exactly k heads in n tosses.

Let us assume p is the probability of obtaining a head in any toss. We also assume that these tosses are independent. So, let us define a random variable x that represents the number of heads that we obtain in these n coin tosses. Now the sample space for x , you can verify goes from takes the value 0 or 1 or n . 0 represents that you observe 0 heads in all the n tosses 1 represents we exactly observe 1 head in n tosses and n represents that all the tosses results in a head.

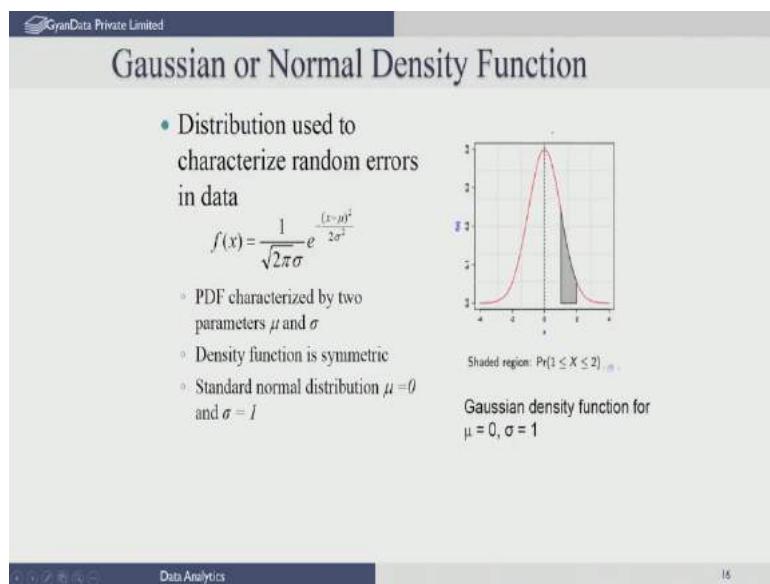
So, clearly each of these outcomes have a certain probability which we can compute and this probability can be computed as follows: let us take the case of k heads in n tosses. Now let us look at one outcome which results in such a such a event. Here I listed the first k tosses as resulting in a head and the remaining $n - k$ tosses resulting in a tail.

Clearly the probability of this particular event is p^k , because I am getting p successive heads and these out tosses are independent. So, p^k is the probability that you will get k successive heads and $1 - p^{(n - k)}$ would represents the probability that you will get $n - k$ successive tails. So, $p^k (1 - p^{(n - k)})$ represents the probability of this outcome. However, this is only one such outcome of obtaining n heads. You can rearrange these heads and it is equal into picking k heads out of n tosses. The number of ways in which you can pick k heads out of n tosses is defined by this combination $n!$ divided by $k!$ into $(n - k)!$.

So, the probability finally, of receiving k heads in n tosses which is denoted by f the random variable taking the value k is given by the probability, which is defined on the right hand side here, this distribution for various values of k. For example, x = 0, x = one can be computed and such a distribution will be called as the probability mass function. For this particular random variable and this particular distribution is called a binomial distribution. As an example of the binomial distribution mass function is shown on the right hand side for n = 20 and taking the probability p = 0.5 clearly, it shows the probability of receiving 0 heads in 20 tosses is extremely small and similarly the probability of obtaining 20 heads in 20 tosses loss is also small as expected the probability of obtaining ten heads has the maximum value as shown.

The vertical line here represents the probability value for a particular number of heads being observed in n tosses. So, clearly since it is a fair coin, we should expect that out of the 20 tosses, 10 heads are most likely to be obtained and this has the highest probability. This distribution is characterized by a parameter p a single parameter p and we can also see for large enough n it tends to this bell shaped curve which is the what is called the Gaussian distribution , which will make use of for large n instead of using the binomial distribution which is computationally more difficult we can approximate by a Gaussian distribution for computational purposes.

(Refer Slide Time: 10:09)



That is an example of a probability mass function of a discrete random variable.

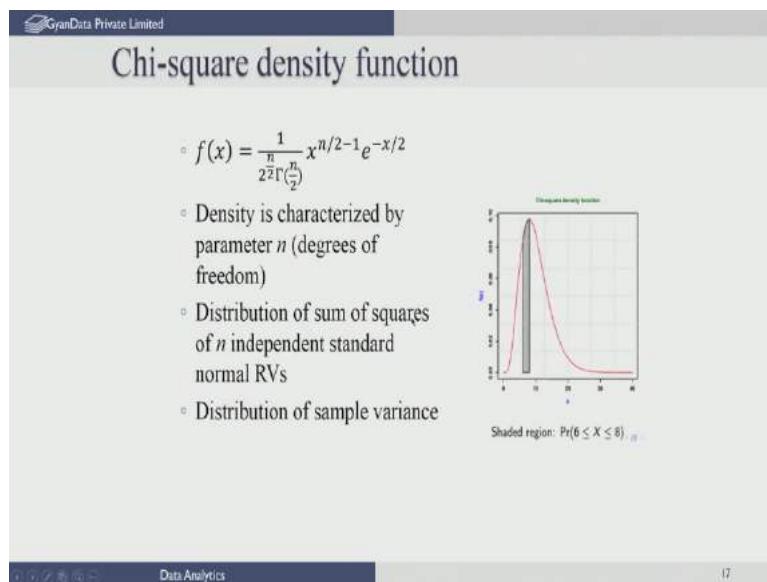
We have now considered a continuous random variable. In this case we will look at what is called the normal density function which is shown on the right. Usually this normal density function is used to characterize what we call random errors in data and it has this density function as given by this

$$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Now, this particular density function has two parameters μ and σ and it has the shape as shown here like a bell shaped curve, which is the normal density function. Notice that it is symmetric and in a particular case of this normal density function is when $\mu = 0$ and $\sigma = 1$ and such a normal density function with mean $\mu = 0$ and $\sigma = 1$ is called a standard normal distribution.

Again, if you want to compute the probability of this, that the standard normal variable lies within some interval. Let us say 1 and 2 you have to compute the shaded region. Unfortunately, you cannot analytically do this integration of this function between 1 and 2 you have to use numerical procedures and the our package contains, such functions for computing the probability numerically such that the variable lies within some given interval we will see such computations a little later.

(Refer Slide Time: 11:37)



Another continuous random variable who is characterized by density function known as the χ^2 square density function. We do not need to remember the form of this function it has them γ function and so, on. Again this density function is characterized by one parameter which is n called the degrees of freedom. Notice that this function takes values only between the random variable which follows this distribution takes values only between 0 and ∞ . The probability to have

negative values is defined to be exactly = 0 and it turns out that this distribution arises when you square a standard normal variable.

So, if you see, the square of a standard normal variable will be a χ^2 square distribution with one degree of freedom. If you take n independent standard normal variables and square and add all of them that will result in a χ^2 square distribution with n degrees of freedom, n representing the number of standard normal variables which you have squared and added to get this new random variable. We can show later that the distribution for sample variance follows a χ^2 square distribution and therefore, it is used to make inferences about sample variances.

(Refer Slide Time: 12:59)

Moments of a pdf

- Similar to describing a function using derivatives, a pdf can be described by its moments
 - For continuous distributions
 - $E[x^k] = \int_{-\infty}^{\infty} x^k f(x) dx$
 - For discrete distributions
 - $E[x^k] = \sum_{i=1}^N x_i^k p(x_i)$
- Mean : $\mu = E[x]$
- Variance : $\sigma^2 = E[(x - \mu)^2] = E[x^2] - \mu^2$
- Standard deviation = Square root of variance = σ

There are other examples of probability density functions such as the uniform density function and exponential density function which we have not touched upon. I would T distribution and. So, on which you can actually look up what we want to do is describe some properties of these density functions.

Just as you take a function and talk about properties such as derivatives we can talk about moments of a probability density function. And these moments are described by what is called the symbol expectation e of some function. So, in this particular case, I have taken the function to be x power k and expectation of x power k $E[x^k] = \int_{-\infty}^{\infty} x^k f(x) dx$. This is called the moments of the distribution. If k = 1, you will call it the first moment. If k = 2 you will call the second moment and so on so, forth. So, if you give all the moments of a distribution it is equivalent to stipulating the density function completely.

Typically we will usually specify only 1 or 2 or 3 moments of the distribution and work with them for discrete distributions. You can similarly describe this moment; in this case expectation of x power k is defined as summation -integrations replaced by summation- over all possible outcomes of the random variable x . I represent the outcome and k is the power to which you have raised. So, x_i power k probability of obtaining the outcome x_i which is very similar to this integration procedure except that the finite number of outcomes and therefore, we have replaced the probability $f(x)dx$ with $p(x_i)$ and the value x^k with the outcome x_i^k and integral as a summation.

Now there are two important moments that we described for a distribution. What is called the mean or the first moment which is defined as expectation of x and the variance which is defined as denoted by the symbol σ^2 and this is defined as the expectation of x - the first moment which is μ whole square.

This is the function that we want to take the expectation of, which can be obtained by $x - \mu$ the whole square $f(x) dx$. In the case of a continuous distribution we can show that the variance σ^2 is expectation of x squared which is the second moment of the distribution about $0 - \mu$ squared. This proof is left to you, you can actually try to prove this; the standard deviation is defined as the square root of the variance.

(Refer Slide Time: 15:50)

The slide has a header 'Properties of Gaussian RVs'. Below the header is a bulleted list of properties:

- For a Gaussian RV x
 - Mean : $E[x] = \mu$
 - Variance : $E[(x - \mu)^2] = \sigma^2$
 - Symbolically $x \sim \mathcal{N}(\mu, \sigma^2)$
- Standard Gaussian RV $z \sim \mathcal{N}(0,1)$
- If $x \sim \mathcal{N}(\mu, \sigma^2)$ and $y = ax + b$ then
 - $y \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$
- Standardization
 - If $x \sim \mathcal{N}(\mu, \sigma^2)$, then $z = \frac{(x-\mu)}{\sigma} \sim \mathcal{N}(0,1)$

Now, for a Gaussian random variable, if I take the expectation of x which is the mean, we can show that is the first parameter μ in the density function and the variance which is denoted defined as expectation of $x - \mu$ the whole squared turns out to be $= \sigma^2$ which is the second parameter of the distribution.

So, the parameters that we have used to characterize the normal variable is μ the mean and σ^2 which is the variance. Typically σ^2 tells you how wide the distribution will be and μ tells me what the value is at which the density function attains the highest probability most probable value. So, μ is also known as the centrality parameter and σ^2 is essentially the width of the distribution tells you how far the values are spread around the central value μ symbolically. We defined this normal distribution variable random variable as distributed as N represents the normal distribution and the parameters are defined by μ and σ^2 which completely defines this density function. A standard normal or standard Gaussian random variable is a particular random variable, which has normally distributed random variable which has mean 0 and standard deviation one and denoted by this symbol.

Now, we can show that if x is a normally distributed random variable with mean μ and σ^2 , then if you take a linear function of this x denoted by $ax + b$, where a and b are some constants, then we can show that y you will also have a normal distribution. But its mean will be our expected value will be $a\mu + b$ and its variance would be a squared σ square.

Now we can use this linear transformation to do something called standardization, which you will see often in many many application of hypothesis testing or estimation of parameters, where we simple define if a random variable is normally distributed with mean μ and σ^2 variance. Then we can de ne a new random variable z which is $x - \mu$ by σ . That is, you subtract the mean from the variable random variable and divide it by the standard deviation that, this new random variable is a linear transformation or a linear function of x and therefore, we can apply the previous rule to show that z is a standard normal distributed variable which means it has a mean 0 and a standard deviation of variance = 1, this is this process is also known as standardization.

(Refer Slide Time: 18:39)

The screenshot shows a presentation slide with a dark blue header containing the GyanData Private Limited logo. The main title is "Computation of Probability using R". Below the title is a bulleted list of points:

- Function to compute probability given a value X
- Lower tail probability = $P(-\infty < x < X) = \int_{-\infty}^X f(x)dx$
- Functions `pnorm(X, mean, std, 'lower.tail' = TRUE/FALSE)`
 - *norm* refers to the distribution and can be replaced by other distributions (chisq, exp, unif)
 - X is the value (limit)
 - Parameters of the distribution (eg. mean and std for normal distribution)
 - `lower.tail = TRUE` (default) to obtain lower tail probability and `FALSE` to obtain upper tail probability

At the bottom of the slide, there is a dark blue footer bar with the text "Data Analytics" on the left and the number "20" on the right.

Now in our there are several functions that allow you to compute probability given a value or the value given the probability. So, we will see some couple of examples of such functions. For example, if you give a value x and ask what is the probability that this continuous random variable lies between the interval $-\infty$ to this capital x value that you are given then, obviously, I have to perform this integral $-\infty$ to x of the density function.

And as I said this can only be done numerically and there is a function for doing this it is called `pxxx`. In the case of a normal distribution you call it `pnorm` you give it the value upper limit in this case and then you define the mean and standard deviation of the two parameters of the normal distribution and you also specify something; whether you want the upper tail probability or the lower tail probability. We will see what it is and this function will give you the probability value.

This value of this integral, notice this integral is nothing but the area under the curve between $-\infty$ to x, if `lower.tail = TRUE` it will give you this integral value area between $-\infty$ and x. On the other hand if `lower.tail` is `FALSE`, then it will give the in area under the curve between x n ∞ . So, x will be taken as the lower limit and ∞ is the higher limit if you say `lower.tail = TRUE` it will take excess the upper limit and do the area in under the curve $f(x)$ between $-\infty$ and x the default value is `TRUE`. So, this norm can be replaced by other distributions like χ square exponent uniform and so on, so forth to give the probability for other distribution given this value x.

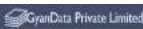
Now, the parameters of the distribution must also be specified for every case in the normal distribution. There is only there are two parameters, but other distributions such as χ squared may have one parameter such as the degrees of freedom and exponent will have one parameter such as the parameter λ and so on so forth. As I said the lower dot tail tells you whether you want the area to the left of x or to the right of x .

(Refer Slide Time: 21:00)

Then other functions in R one of them is qnorm which actually does what is called the inverse probability; here you give the probability and ask what is the limit X. So, here I have if you give the probability to q norm with the mean and standard deviation parameters of the normal distribution and you say the lower tail = 2, then it will actually compute the value of X such that the integral between $-\infty$ to X of this density function = the given value p you are specified p and calculating X.

In p norm you are specifying x and computing p. So, this is called the inverse probability function as before if you actually say lower dot trail = FALSE, then this integral will be replaced by x to ∞ such that x to ∞ of f(x) dx = p and it will find the value of x. There are other functions called d norm which computes the density function value at a given x and r norm which are used to generate random numbers from this given distribution.

(Refer Slide Time: 22:04)

 GyanData Private Limited

Joint pdf of two RVs

- Joint pdf of two RVs x and y: $f(x,y)$
- $P(x \leq a, y \leq b) = \int_{-\infty}^b \int_{-\infty}^a f(x,y) dx dy$
- Covariance between x and y: $\sigma_{xy} = E[(x - \mu_x)(y - \mu_y)]$
- Correlation between x and y: $\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$
- Two RVs x and y are uncorrelated if $\sigma_{xy} = 0$
- Two RVs x and y are independent if $f(x,y) = f(x)f(y)$



Now, having seen the distribution of single random variable, let us move on to the joint distribution of two random variables. In general, we will be dealing with the multivariate distributions which are joint distribution of several random variables, but first we will look at the joint probability density function of two continuous random variables x and y denoted by the function $f(x,y)$.

And the way this density function is used to compute the probability is as before the joint probability that $x \leq a$ - ∞ being the other assumed to be the other limit and $y \leq b$ or y ranging from $-\infty$ to b , the joint probability of these two variables lying in these intervals is denoted as computed as the integral $-\infty$ to b and double integral $-\infty$ to a $f(x) y dx dy$.

That is the basically the volume of this particular function $f(x)$ comma y . Now when there are two variables you can also define other than the variance of x and y which are denoted as σ^2_x and σ^2_y . You can also define what is called the covariance between x and y and this is defined as the expectation of $x - \mu_x$, μ being the mean or expectation of x multiplied by $y - \text{expectation of } y$ which is denoted by μ_y this product is the expectation of this product function is defined as the covariance between x and y and denoted by the symbol σ_{xy} .

Now, the correlation between x and y is the standardized or normalized form of this covariance which is nothing but σ_{xy} divided by the standard deviation of x and the standard deviation of y -this is denoted by the symbol ρ_{xy} . We can show that ρ_{xy} varies between -1 to $+1$ depending on the extent of correlation between x and y . Typically, when $\sigma_{xy} = 0$, we $\rho_{xy} = 0$ and we say that the random variables x and y are uncorrelated.

On the other hand, if x and y are independent, then the joint density function of $f(x)$ x comma y can be written as the product of the individual density functions or marginal density functions of x and y . That is $f(x)$ into f of y . This is the extension of this notion of independent variables in terms of probability we defined in the previous lecture where we said the probability joint probability of x and y is basically probability of x into probability of y .

But this is a generalization which defines the notion of independence of two random variables.

(Refer Slide Time: 25:01)

- A vector of RVs $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T$
- Multivariate Gaussian Distribution ; $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
 - $E[\mathbf{x}] = \boldsymbol{\mu}$: Mean vector
 - $E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T] = \boldsymbol{\Sigma}$: Variance-covariance matrix
 - $f(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}$: pdf

- Structure of $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} & \cdots & \sigma_{x_1 x_n} \\ \sigma_{x_2 x_1} & \sigma_{x_2}^2 & \cdots & \\ \vdots & \vdots & \vdots & \\ \sigma_{x_n x_1} & \cdots & \cdots & \sigma_{x_n}^2 \end{bmatrix}$$

Now, we can now extend this idea of joint distribution of two variables to joint distribution of n variables. Here I have defined the vector \mathbf{x} which consists of n random variables x_1 one to x_n . And specifically we look at this multivariate normal distribution we denoted by the symbol \mathbf{x} multivariate normal with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\sigma}$. Now each of these x_i components x_1, x_2 and so on have their respective means. If you put them in the vector form, we get this mean vector symbolically written as expectation of \mathbf{x} which is a multi dimensional integral, we get this value $\boldsymbol{\mu}$ which is known as the mean vector. And similar to the variance we defined what is called the covariance matrix which is defined as expectation of \mathbf{x} about the mean $\boldsymbol{\mu}$ or $\mathbf{x} - \boldsymbol{\mu}$ into $(\mathbf{x} - \boldsymbol{\mu})^T$. Remember this is a matrix of variables because \mathbf{x} is a vector and if you take the expectation of each of these elements of this matrix you will get this matrix called the variance covariance matrix.

In fact, we can write the multi dimensional normal distribution also has a very similar form if you look at it $1/2\pi$ we had square root of 2π . In this case it will be $2\pi^n$ n by 2 n represents number of dimension of this vector and we had $\boldsymbol{\sigma}$ in this case. We have the square root of this matrix covariance matrix $\boldsymbol{\sigma}$ and we had exponents - half we had a quadratic form. In this case the quadratic form is $(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$ which is similar to $(\mathbf{x} - \boldsymbol{\mu})^2$ divided by σ^2 .

So, it has very similar form we do not need to know the form. We need to know how to interpret $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. And if you look at the structure of $\boldsymbol{\sigma}$ you will find that it is a square matrix with the diagonally elements representing the variance of each of the elements that is $\sigma_{x_1}^2$ is the variance of x_1 and $\sigma_{x_2}^2$, the variance of x_1 and so on, so forth. And the off diagonal elements representing the covariance between x_1 and x_2 or x_1 and x_3 , x_2 and x_3 and so on ,so forth pair wise covariance.

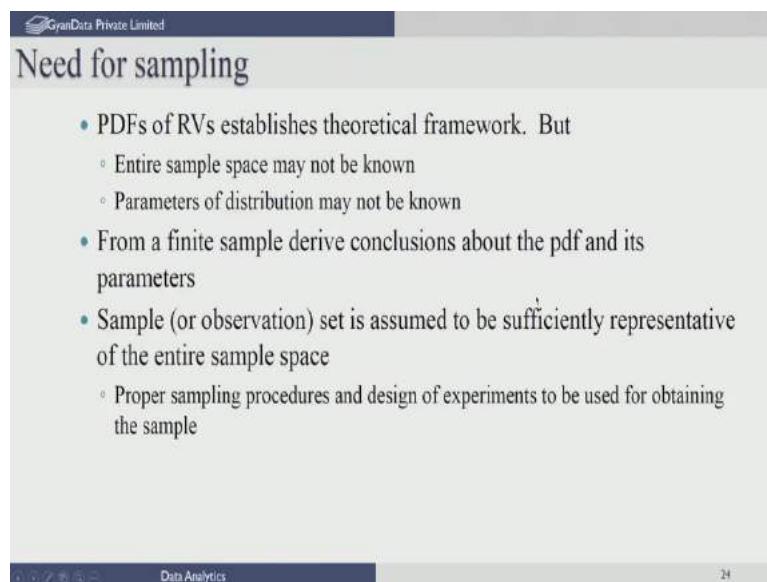
Those are the off-diagonal elements for example, σ_{x_1, x_2} represents the covariance between x_1 and x_2 , σ_{x_1, x_n} represents the covariance between x_1 and x_n . This particular matrix is symmetric and we completely characterize the multivariate normal distribution by specifying this mean vector μ and the covariance matrix σ .

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 21
Sample Statistics

In the preceding two lectures, I introduced the concepts of probability. Probability provides a theoretical framework for providing, for performing statistical analysis of data. Statistics actually deals with the analysis of experimental observations that we have obtained. So, in this lecture I will introduce you to a few measures statistical measures and how they are used in analysis.

(Refer Slide Time: 00:47)



GyanData Private Limited

Need for sampling

- PDFs of RVs establishes theoretical framework. But
 - Entire sample space may not be known
 - Parameters of distribution may not be known
- From a finite sample derive conclusions about the pdf and its parameters
- Sample (or observation) set is assumed to be sufficiently representative of the entire sample space
 - Proper sampling procedures and design of experiments to be used for obtaining the sample

Data Analytics 24

So, what is the need for performing statistical analysis when we have already talked about probability density functions and so on?

Typically, when we are actually doing analysis we do not know the entire sample space. We may also not know all the parameters of the distribution from which the samples are being withdrawn. Typically we actually obtain only a few samples of the total number of available population. So, from this finite sample we have to derive conclusions about the probability density function of the entire population and also infer, make inferences about the parameters of these distributions.

So, the sample or observation set is supposed to be sufficiently representative of the entire sample space. Let us take an example.

Suppose you want to actually find out the average height of people in the world, you cannot go and sample just people or take heights of American people alone because they are known to be much taller compared to the Asian people.

So, when you take samples you should take examples from let us say America, from Europe from Asia and so on, so forth. So that you get a representative of the entire population of this world. So, this is called proper sampling procedures and these are dealt with in the design of experiments. We will assume that we have obtained a sample; you have done the due diligence and obtained the representative sample of whatever population you are trying to analyze.

(Refer Slide Time: 02:21)

The slide has a dark blue header with the GranData Private Limited logo. The main title 'Basic Concepts' is in bold black font. Below it is a bulleted list:

- Population: Set of all possible outcomes of a random experiment characterized by $f(x)$
- Sample set (realization) : Finite set of observations obtained through an experiment
- Inference: Conclusion derived regarding the population (pdf, parameters) from the sample set
 - Inference made from a sample set is also uncertain since it depends on the sample set which is one of many possible realizations

A small video thumbnail of a man speaking is visible in the bottom right corner of the slide area.

Now, with basic definition of population is the set of all possible outcomes of this random expire experiment. We have already defined this as the sample space. What you are going to obtain is just a few examples or samples and these are called the sample set. Its an infinite set of observations obtained through whatever experiment that you are going to conduct. Now from this sample set you want to make inferences which are conclusions that you derive regarding the population itself, which you do not know its either the probability density function of the population or the parameters of the population.

You have to note that when you actually lose such inferences, your inference is also stochastic or uncertain because the sample that you have drawn are also uncertain; they are not representative of the entire population. So, you should expect that your inferences are also uncertain and therefore, when you provide the answers, you should

actually provide also the confidence interval associated with these estimates that you have deriving.

So, that is one of the reasons that we actually studied probability density functions because then you can characterize all the estimates that you have obtained from the sample in terms of this confidence interval and so on or the probability that you will obtain this value.

(Refer Slide Time: 03:40)

- Descriptive Statistics (Analysis)
 - Graphical : Organizing and presenting the data (eg. box plots, probability plots)
 - Numerical: Summarizing the sample set (eg. mean, mode, range, variance, moments)
- Inferential
 - Estimation: Estimate parameters of the pdf along with its confidence region
 - Hypotheses testing: Making judgements about $f(x)$ and its parameters

So, let us actually look at some typical analysis statistical analysis. We can divide the statistical analysis into two parts, the graphical part or graphical analysis where we use plots and graphs in order to have a visual feel of the entire data. The other way of doing is to actually do quantitative computations or numerical computations, where you try to summarize the entire sample sent by a few parameters. Example we will talk about mean and variance and so on. Notice that we have taken hundreds of data points or experiments, you cannot go and tell somebody all the hundred values, you cannot reel off all these values that will not be possible for somebody to digest.

Summary statistics that we do numerically allows you to get a feel for the entire data set that you have obtained without knowing the individual observations. And that is why these are very useful and they are also called summary statistics. Now inferential statistics deals with two kinds of problem estimation problem, where we try to estimate parameters of the probability density function.

We did talk about parameters such as the expected value or the first moment and the second moment and so on, and different distributions are different number of parameters and how do we estimate these parameters from a small sample that we obtain. And how do you give a confidence region for these estimates that is called estimation and the

other kind of decision making that we want to do is; we want to judge whether particular value is 0 or not. The parameters of the distribution and such decision making that we do from a sample is called hypothesis testing.

We want to know whether a customer will continue to remain with you or will leave you for another vendor, based on whatever offers that you are making. So, these are come under the category of hypothesis testing. We will first deal with the descriptive statistics and in the next lecture we will deal with inferential statistics.

(Refer Slide Time: 05:46)

Measures of Central Tendency - Mean

- Represent sample set by a single value
 - Mean (or average): $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$
 - Best estimate in least squares criterion
 - Unbiased estimate of population mean: $E[\bar{x}] = \mu$
 - Affected by outliers
 - Eg: Sample heights of 20 cherry trees
[55 55 59 60 63 65 66 67 67 67 71 71 72 73 75 75 78 81 82 83]
 - Mean = 69.25 (population mean used to generate random sample was 70)
 - Mean = 71.75 (after a bias of 50 was added to first sample value)

So, some of the summary statistics that we can define for a sample or what we call measures of central tendency, it is the kind of the center point of this entire sample you might say. And let us define what is called the mean, these are measures that you are familiar with from your high school courses in mathematics.

So, let us recap some of these. The mean of a sample is defined as the summation of all the data points that you obtain divided by the number of datapoints that you have. So, that is also denoted by the symbol \bar{x} and its also called the mean or the average of the sample. This particular thing as I said can be viewed as a central point of the entire sample that you have got.

And we can show that this estimate that we obtained of the sample the average is the best estimate in some sense. Later on, we will set up what is called the least squares method of estimating parameters. And if you set up this particular criterion for estimating parameters you will find that \bar{x} is the best estimate that you can get from the given sample of data. We can also show some properties of this estimate. For

example, we can prove that this \bar{x} represents an unbiased estimate of the population mean μ which you do not know anything about.

What do we mean by the unbiased estimate? Expectation of \bar{x} is μ . This can be analytically proven for any kind of distribution. And in order to prove understand what this means you say that suppose you take a sample of N points and you get an estimate \bar{x} . And you repeat this experiment and draw another random sample from the population of N points and get another value of \bar{x} .

And you average all these \bar{x} s that you get from different experimental sets, then the average of these averages will tend to this population mean. That is a way of interpreting this statement that its an unbiased estimate. There are other properties of estimates we will see that we want the demand, but this is an useful and important property of estimates that you should always check.

The one unfortunate aspect of this particular statistic or mean is that it is if there is one bad data point in your sample, by mistake you have made a wrong entry, then your estimate of \bar{x} can be significantly affected by this bad value. The bad value is what we call an outlier and even a single outlier in your data can give rise to a bad estimate of \bar{x} .

Let us take one example we have taken 20 cherry trees and we have measured the heights of the cherry trees in terms in feet and we got let us say the set of bunch of numbers; generated these randomly from a normal distribution with some mean which is 70 and the standard deviation of 10. So, the population mean is 70 and the population standard deviation is 10 and I got these values. You can use `rrnorm` for example, in `r` in order to generate such data points.

Now, if you take this sample of 20 points and compute the mean, you get a value of 69.25 which is very close to the population mean. So, you see that it is a good estimate of the population mean, even though you did not know what that value was until I told you. Now on the other hand if I take the first data point 55 and add a bias wrongly enter it as 105 let us say by adding 50 to it and then recompute this mean, I will find that the mean becomes 71.75.

It starts deviating from 70 you see more significantly. A single bias in this sample actually caused your estimate to become poorer. That is what we mean by saying that \bar{x} will get affected by outliers in the data. We can define other measures of central tendency which are robust with respect to the outliers, even if the outlier exists, it does not change by much and we will see what that such a measure is.

(Refer Slide Time: 10:13)

• Represent sample set by a single value

- Median: Value of x_i such that 50% of the values are less than x_i and 50% of observations are greater than x_i
 - Robust with respect to outliers in data
 - Best estimate in least absolute deviation sense
 - Eg: Sample heights of 20 cherry trees
[55 55 59 60 63 65 66 67 67 71 71 72 73 75 75 78 81 82 83]
 - Median = 69 (population mean used to generate random sample was 70)
 - Median = 69 (after a bias of 50 was added to first sample value)

Another measure of central tendency is what is called a median. The median is a value such that 50 percent of the data points lie below this value and 50 percent of the experimental observations are greater than this value. So, you like to find out that value below which half the data points lie and above which half the data points lie. For doing this you have to order all the observations that you have got from smallest to highest and then find out the middle value. Let us do this through an example.

So, same 20 cherry trees data I have looked at, this data point I have ordered from the smallest to the largest. And if you look at it the tenth point 1, 2, 3, 4, 5, 6, 7, 8, 9, 10; tenth point is 67 because there are even number of points, the eleventh point is 71 and you take the average between this and call that the median. If there are odd number of points then you can take the middle point just as it is because there are even number of points, you take the average of the mid midpoints, in this case the tenth and the eleventh point and that gives you a median of 69.

Suppose we add a bias in the first data point as before and make this 105 and then reorder the data and find out the again the median; we find that the median has not changed.

So, the presence of an outlier in this particular case has not affected the median at all and that is why we call this a robust measure even if there is a bad data point in your samples. You can also show that this estimate is the best estimate in some sense. In this case the merit that you are using is what is called the absolute deviation. That is you are asking what is the estimate which deviates from the individual observations in the absolute sense to the least extent? And it turns out that the median is such a point, such an estimate. So, when there are

outliers typically we would like to use this as a central measure rather than the mean.

(Refer Slide Time: 12:25)

The screenshot shows a presentation slide from 'GyanData Private Limited' titled 'Measures of Central Tendency -Mode'. The slide content includes:

- Represent sample set by a single value
- Mode: Value that occurs most often (Most probable value)
- eg. Sample heights of 20 cherry trees

[55 55 59 60 63 65 66 67 67 67 71 71 72 73 75 75 78 81 82 83]

◦ Mode: 67 (three occurrences)

A video player interface is visible at the bottom, showing a thumbnail of a man speaking. The video player controls include arrows for navigation, a play/pause button, and a volume icon. The word 'Data Analytics' is visible at the bottom left of the slide area.

A mode is another measure of central tendency and this value is the value that occurs most often or what is called the most probable value. And if you take the example of this 20 cherry trees data, we find that the most probable value, the value that repeats more often, is 67.

Again you said this is 3 consecutive, 3 occurrences of this as compared to any other data point and that is called the mode. And in a distribution if it is a continuous distribution, this represents the highest value of this maximum value of the density function. And you should expect most of the data to be clustered around this most probable value. Sometimes distribution may have two modes. What is called a bimodal distribution in which case if you sample from such a distribution, you will find two clusters one clusters around the one of the modes and another cluster around the second mode. So, you should interpret the mode as that value around which you will find most of the data points.

(Refer Slide Time: 13:35)

Measures of Spread

- Represents spread of sample set
 - Sample variance : $s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$
 - Unbiased estimate of population variance : $E[s^2] = \sigma^2$
 - Standard deviation is sqrt of variance
 - Mean absolute deviation : $\bar{d} = \frac{1}{N} \sum_{i=1}^N |x_i - \bar{x}|$
 - Range : $R = x_{max} - x_{min}$
 - Eg. Sample heights of 20 cherry trees
 $s^2 = 70.5132$ and **212.25 with outlier**
 $s = 8.392$ (population std used for generating numbers was 10)
 $MAD = 6.85$ and **9.5 with outlier**
 $Range = 83 - 55 = 28$

The another measure which characterizes a sample set is what we call the measures of spread and tells you how widely their data is ranging. So, one of the measures is what to call the sample variance denoted by the symbol s squared and that is defined as the data point x_i - the sample average that you have already computed. This deviation of the observation from the sample mean u square.

And add over all the data points n data points and divide the this particular sum squared value by $N - 1$, such a measure is called the sample variance. And again you can prove that the sample variance is an unbiased estimated estimate of the population variance and the square root of the sample variance is also known as the standard deviation.

Now, just like the mean, the sample variance happens to be also a very susceptible to outliers. So, if you have a single outlier, the sample variance, our sample standard deviation can become very poor estimate of the population parameter. So, we define another measure of spread which is called the mean absolute deviation somewhat similar to the median. In this case instead of taking the sum squared deviation, we take the absolute deviation of the data point from the mean; you can also take it from the median if you wish. So, deviation of the observation from the mean or the median, you take the absolute value of this deviation sum over all the end points and divide by N and that is what is called the mean absolute deviation.

Again whether you should divide by N or $N - 1$ is a point to be noted. Typically we divide by $N - 1$ to indicate that if you have only one data point; it is not possible to estimate s^2 . For example, if

you have one data point, s^2 will turn out to be 0 because the mean will be equal to the point. So, really speaking you have only $N - 1$ data points to estimate the spread. So, that is why we divide by $N - 1$ to indicate that one data point has been used up to estimate the sample mean or the median whatever the parameter that you are actually estimated.

So, similarly here also you can divide by $N - 1$ to indicate that only $N - 1$ data points were available for obtaining the mean absolute deviation. A third measure of spread is what is called the range that is basically the difference between the maximum and minimum value. All of these give you indication of how much the data is spread around the central measure which is the mean or the mode or the median as the case may be.

So, let us take an example of the 20 cherry trees we have computed the variance from the given data. And we find out that its actually 70.5 5132 and if we take the standard deviation; we will find its 8.4. As I told you that I had used a standard deviation of 10 to generate this data from a normal distribution. And we find that the sample standard deviation is a reasonably good measure of the population parameter which we did which was unknown.

On the other hand, if I add outlier of 50 units to the first data point and recompute s^2 and s , it turns out s^2 turns out to be 212 and you can see if I take the square root it might be around 14, which is significantly deviating from the population parameter 10. So, a single outlier can cause the standard deviation and the variance to become very poor and therefore cannot be trusted as a good estimate of the population standard deviation or variance.

On the other hand, let us look at the mean absolute deviation. In this case I have, if we do not have an outlier, we get a mean absolute deviation of 6.9, which is not too bad compared to 10. The moment you have an outlier, the mean absolute deviation shifts to 9.5, it comes closer that is not what is important, but it does not change much just because of the presence of the outlier. So, this is a much more robust measure. In fact, if you take the mean absolute deviation from the median, it would be even better in terms of robustness with respect to the outlier. The range of the data can be obtained as the maximum and minimum value and I have just simply reported it.

So, these are measures of spread. So, even if I do not give you the entire 20 data points and I tell you the mean is, let us say 69 and the standard deviation is 8.5, then you can say that the data will spread typically between $69 +/- 2$ times the standard deviation which is 16. So, the lowest value will be about 53 and the highest value will be about 85 and it turns out if you look at the highest and maximum value and that is what it turns out to be.

So, + or - 2 times the standard deviation from the mean would represent about 95 percent of the data points if the distribution is normal. For other distributions you can derive these kind of intervals if you wish, but just giving two numbers allows me to tell you some properties of the sample and that is the power of these sample statistics.

(Refer Slide Time: 19:16)

- Sample mean
 - For any distribution sample mean is an unbiased estimate of population mean
 - If $x_i \sim \mathcal{N}(\mu, \sigma^2)$ and all observations are mutually independent, then $\bar{x} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{N}\right)$
- Sample Variance
 - For any distribution sample variance is an unbiased estimate of the population variance
 - If $x_i \sim \mathcal{N}(\mu, \sigma^2)$ and all observations are mutually independent, then $\frac{(N-1)s^2}{\sigma^2} \sim \chi^2_{N-1}$

Now, there are some important properties of the sample mean and variance which we will use in hypothesis testing. So, I want to recap some of these. If you have observations drawn from the normal distribution with some population parameter μ and population variance σ^2 . And let us say you draw N capital N observations for all from this distribution; let us assume these draws or samples that you are drawn are independent, it does not have a bias in any manner.

And if you compute the sample average from this set of samples independent samples, then you can prove that \bar{x} is also normally distributed with the same mean population mean μ . Which means the expected value of \bar{x} is μ as I told you before and the expected variance of \bar{x} however, is σ^2/N .

So, the variance of \bar{x} is actually lower than the variance of the individual observations. The important point here to be noted is, if I have N repeats from the same distribution and if I take the average of them, the average will be less noisy than the original observations.

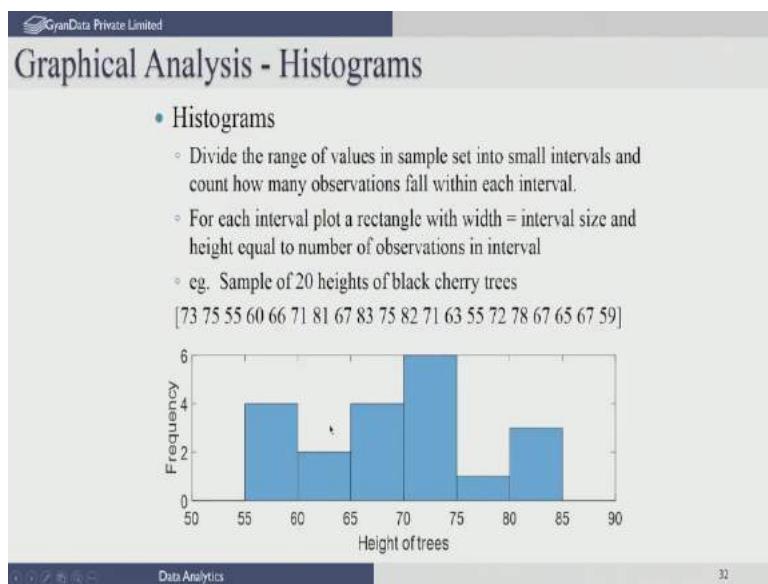
So, one simple way of dealing with noise and reducing the noise content in observations is to take n observations at the same experimental condition and average them. The average will contain less variability or less noise and it will reduce the variance of this

average will be 1 by N times the variance in your individual observations. So, what we call the noise will be reduced by square root of N where N is the number of samples.

Now, if you look at the sample variance and want to characterize the distribution of the sample variance, we can show again if you draw samples from the normal distribution with some mean μ and variance σ^2 and these observations I am going to assume are mutually independent.

Then if you take $N - 1$ times the sample variance divided by the population variance, we can show that this particular measure is a χ^2 squared distribution with $N - 1$ degrees of freedom. We already saw the χ^2 squared is a distribution of a random variable which varies between 0 and ∞ . And that distribution can be used to characterize s^2 and we can later on do hypothesis testing, whether the σ^2 is some value and so on, using these distributions we will see.

(Refer Slide Time: 22:08)



Now, those are numerical methods of actually doing analysis of a sample data. What we want to do is also graphical analysis this is something that you should always do when you are given a data set. The first and foremost that you should do is to do some plotting to get a visual appeal because the mind is capable of inferring things what numbers do not tell you.

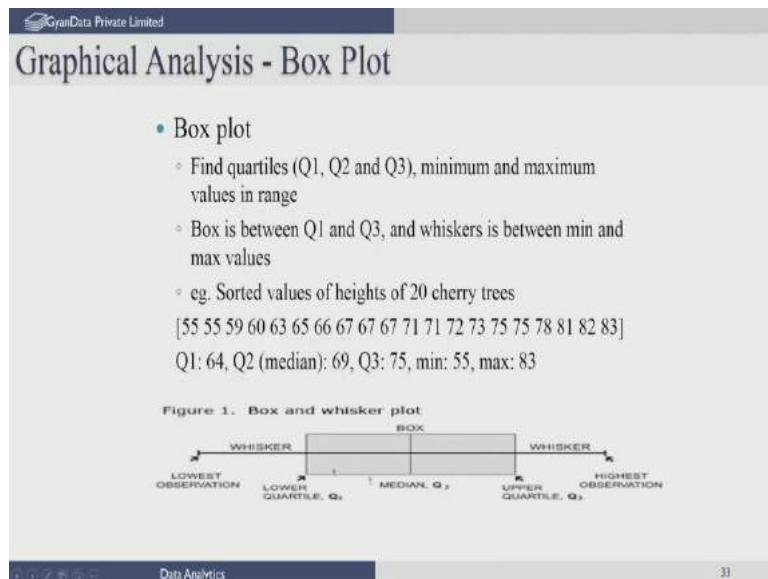
So, my suggestion is always when you have a data set, if you can plot and visualize it please do. So, let us see some of the standard plots again. Some of it you might have already encountered in your high school days. We will start with what is known as the histogram. Here I am given a sample set and what we do is first divide this sample set into small ranges; we define a small range and count how many

observations fall within that range or within each interval. And then we plot the width of the interval or the interval size of the x axis and the number of data points we see in that interval as the y axis, we call it the frequency and that is on the y axis.

So, let us take this example of the cherry trees. We have 20 data points. What I did was divided into small intervals of 5 feet which means I asked what are the number of cherry tree heights which are falling in the range 50 to 55, 55 to 60, 60 to 65 and so on, so forth. And I find between 50 and 55 there are no trees within that height, we find 4 trees with the height between 55 and 60 which we can easily see there is one data point here, there is second data point here, there is a third data point here and fourth data point is 60; so, the edge.

So, the 4 data points lying between 55 and 60 and similarly we find there are two data points lying just above 60 and up to 65 and so on, so forth. And that is what we plotted as a rectangle for each interval and this is known as a histogram. In fact, if I take 100 such examples and I plot, you will find this standard bell shaped curve. And that is because I drew these samples from the normal distribution. In this case because it is 20, you are not able to clearly see its bell shaped. You can see that the most of the data points are clustered around the middle point which is around 70 and you can see highest number 6 there. So, at least that is borne out.

(Refer Slide Time: 28:48)



You have other kinds of plots. One is called the box plot, which is used most often in sometimes in visualizing stock prices. Here you will compute quantities called quartiles Q_1 , Q_2 and Q_3 and the minimum and maximum values in the range. What are quartiles? Quartiles are

basically an extension of the idea of median. Q2 is exactly the median which means half the number of points fall below the value of Q2 and half the number of points are exactly about Q2.

Similarly, Q1 represents the 25 percent value which means 25 percent of the observations fall below Q1. 75 percent above Q1 and Q3 implies that 75 percent of the data points fall below Q3 and 25 percent above Q3. And once you have these values, the median, the quartiles and the minimum maximum, you can plot what is called the box and whisker plot in the box the median is the center value and the lower quartile Q1 and the upper quartile is also plotted.

And the box is drawn between Q1 and Q3 clearly showing where the median is. In this case its shown as symmetric, but generally need not be, either Q2 might be closer to Q1 or Q3. We also show the minimum and maximum values; here in this case the lowest observation the highest observation and those are called the whiskers. This gives you an idea better idea of the spread of the data than just giving you standard deviation or the mean absolute deviation and so on. This gives you a little more information about the spread of the data.

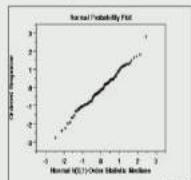
So, as an example if you take the 20 cherry trees and sort them out, sort it from the lowest to highest value, and we try to compute the median it turns out the median is the average of the tenth and eleventh point which is 69. Then the quartile one can actually be computed by just taking the first half which is the 10 points and computing the median of the first 10 points which turns out to be 64. And the Q3 can be computed as the median of the other half from 60 from 71 to 83 and that turns out to be around 75. So, the min and max of course, is 50 and 83 and therefore, you can perform this plot.

(Refer Slide Time: 27:21)

GyanData Private Limited

Graphical Analysis – Probability Plot

- Probability plot (p-p or q-q plot)
 - Determine different quantile values from sample set. Plot computed quantiles vs theoretical quantile values from chosen distribution
 - Same example (standardized and sorted values)
[-1.697 -1.697 -1.2206 -1.1016 -0.7443 -0.5061 -0.3870 -0.2679
-0.2679 0.2084 0.2084 0.3275 0.4466 0.6848 0.6848
1.0420 1.3993 1.5184 1.6374]



Normal Probability Plot

Normal Q-Q (Data Standardized)

Data Analytics

34

The third kind of plot which is very useful is to know about the distribution of the data and this is called the probability plot the p-p plot or the q-q plot. Here instead of determining just Q1, Q2, Q3 you compute several quantiles. And then plot these quantiles against the distribution which you think this data might follow. And if the data falls on the 45 degree line, then you can conclude that the sample data has been drawn from the appropriate distribution you are testing it against.

So, this is useful for visually figuring out from which distribution did the data come from. So, I have taken this example of these 20 cherry trees. I first standardized them, standardization means we remove the mean and divide by the standard deviation and we get the values. The 20 values as these are called the standardized values I have sorted them from the lowest to highest.

Now if you look at the 10 percent quantile, I can say that out of 20 points two points first two points fall below 1.679 - 1.679. So, - 1.679 represents the 10 percent quantile similarly - 1.1016 represents to 20 percent quantile and so on, so forth. For example, the 50 percent quantile or the median quantile, in this case the standardized measure, will be and which is around these two points around let us say - point or around 0 close to 0.

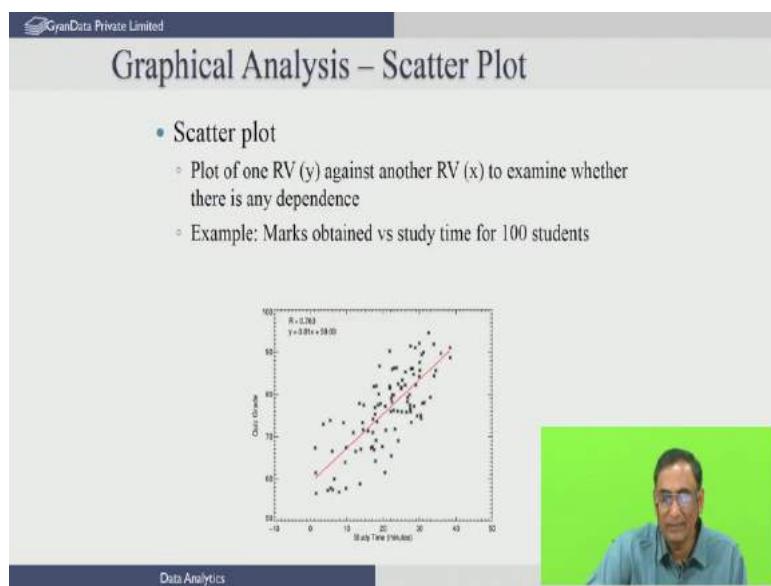
Notice that for a normal distribution, 50 percent of the data will lie below 0 and that is what this also seem to indicate. Now if you go to the standard normal distribution and try to compute the value below which the probability is 0.1 which is the lower tail probability we

talked about. Then you will find the value is around let us say - 1. 7-1.5 whatever that value turns out to be. So, that is the 10 percent quantile. Similarly you ask what is the value below which 20 percent of the data lies or what is the value below which 20 percent of a standard normal distribution values will have a probability of area under the curve of 0.2 and that value you take that is the second 20 percent quantile and so on, so forth.

Then you plot the actual value obtained for the sample which is - 1.67 against the standard normal contact and that is what is called the probability plot. So now, if this data has been drawn from the normal distribution then you should find a curve like this. I did not plug the normal probability plot for these 20 points, but for some other set of data. But typically if you find if you think that this data comes from the normal distribution, then you will find that in the normal probability plot the data will align itself on the 45 degree line and then you can conclude yes that the data has come from the distribution.

You can test this against any distribution. In this case, I have shown you how to test it against the normal distribution. You can take the quantiles from a uniform distribution or from the χ^2 squared distribution what have you and the plot these sample quantiles against the expected population quantiles. And if they fall on the 45 degree line then you know that it comes from the appropriate distribution.

(Refer Slide Time: 30:57)



So, this is useful for determining visually the distribution from which the data have been drawn. Now the last kind of plot which is useful in data analysis is what is called the scatter plot. The scatter plot plots one random variable against another. So, if you have two random

variables, let us say y and x and I want to know whether there is any relationship between y and x , then one way of visually verifying this dependency or interdependency is to plot y versus x .

So, in this case we have taken some data corresponding to 100 students for which I mean students have spent time preparing for a quiz and they have obtained marks in that quiz. So, you should find typically if you spend more time study you should obtain typically more marks. And that is what this is trying to show on the x axis is a time in minutes that we have plotted. And the y axis we have plotted the marks obtained by the student and you can see there is an alignment.

The marks obtained seem to be dependent on the time spent and in fact, in this particular case you find that it looks like a linear dependency. So, you can plot a line approximate line through these data points we will show how to fit such lines using regression and how to obtain the parameters of this line that we have actually indicated here.

But more importantly if the random variable y , in this case the quiz marks has a dependency on the study time, then you will see an alignment of the data. On the other hand if there is no dependency you will find a random spread, this data will be spread all around with no clear pattern, in which case you will say that there are these two variables are more or less independent and you do not have to discover a relationship between these variables.

So, this is a plot which we will do in order to assess dependency between two variables and then proceed for further for analysis. In the next lecture we will take you through some decision making using hypothesis testing and how to perform estimation of parameters using sample data. See you in the next lecture.

Data Science for Engineers
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 22
Hypotheses Testing

Welcome to this last lecture on Introduction to Probability and Statistics. In this lecture we will introduce you to the basics of hypothesis testing which is an important activity when you want to make decision from a set of data.

(Refer Slide Time: 00:39)

GyanData Private Limited

Motivation for Hypotheses Testing

- Business: Will an investment in a mutual fund yield annual returns greater than desired value? (based on past performance of the fund)
- Medical: Is the incidence of diabetes greater among males than females?
- Social: Are women more likely to change mobile service provider than men?
- Engineering: Has the efficiency of the pump (η) decreased from its original value due to aging?

Data Analytics 37

So, to give you some motivation for hypothesis testing we look at some cases. Let us look at a business case where you are interested in investing in a mutual fund and you want to know whether this investment will yield a certain annual return greater than some desired value let us say 15 percent or 20 percent that you might want. Now this decision has to be made based on past performance of the fund which you have data which you can collect.

Similarly, let us assume you are a medical practitioner and you want to ask this question whether the incidence of diabetes is greater among males than females based on data that you have gathered about males and females and what proportion of males and what proportion of

females have diabetes. Now, you can ask similar question in the social sector. You can if you are a service provider mobile service provider you want to know whether women are more likely to change service provider than men and depending on that you might want to provide more incentives accordingly for women to retain them.

In engineering you can ask a question such as a pump efficiency whether it has degraded from its original value due to aging and if so you may want to do some maintenance of the pump. So, these kind of questions that are decisions that you have to make will be based on data that you have gathered about the particular object or item.

(Refer Slide Time: 02:02)

GyanData Private Limited

Hypotheses Testing

- The hypotheses is generally converted to a test of the mean or variance parameter of a population (or differences in means or variances of populations)
- A hypothesis a statement or postulate about the parameters of a distribution (or model)
 - Null hypothesis H_0 : The default or *status quo* postulate that we wish to reject if the sample set provides sufficient evidence (eg. $\eta = \eta_0$)
 - Alternative hypothesis H_1 : The alternative postulate that is accepted if the null hypothesis is rejected (eg. $\eta < \eta_0$)

Data Analytics

38

Now, let us look at how do you perform these decision making using what is called hypothesis testing. Typically what you have to do is to convert this hypothesis into a test for the mean or variance parameter of a population or perhaps a difference in the means of two populations or the difference of variances of the two population.

The statements that previously we saw have to be first converted into a test of a parameter of some population. Now, this hypothesis is a postulate about the parameters. You call them either the null hypothesis or the alternate hypothesis. The null hypothesis, the default hypothesis, that you want to test, the status quo postulate that you wish to reject if the sample set provides sufficient evidence. For example, this is the statement about which you want to make the strongest claim based on the data. So, that you choose as the null hypothesis and what we call the default hypothesis. Example you want you want to convert this statement into a parameter called η , whether this η efficiency, let us say of the pump = η_0 , its original value.

If you have per performance data that you collect then you based on the data you want to reject whether this efficiency at current time is different from its original value η_0 . If there is evidence, you will reject this hypothesis in favor of the alternative hypothesis which is essentially saying that the pump efficiency in this case is less than η_0 . So, this is called the alternate hypothesis. So, you set up the hypothesis such as $\eta = \eta_0$ which you call the null hypothesis and the alternative hypothesis which you want to choose in favor of this if the evidence is there from the data such as for example, η less than η_0 .

So, all hypothesis testing has this null and alternative. And we can have different types depending on whether you are testing for the mean or the variance and whether this is less than or greater than or on both sided and we would see many such examples.

(Refer Slide Time: 04:20)

The slide has a header 'GyanData Private Limited' and a title 'Hypotheses Testing Procedure'. Below the title is a bulleted list of six steps:

- Identify the parameter of interest (mean, variance, proportion) which you wish to test
- Construct the null and alternative hypotheses
- Compute a test statistic which is a function of the sample set of observations
- Derive the distribution of the test statistic under the null hypothesis assumption
- Choose a test criterion (threshold) against which the test statistic is compared to reject/not reject the null hypothesis

At the bottom of the slide are navigation icons and the text 'Data Analytics' and '39'.

Now, in order to perform this hypothesis testing and make a decision; As we said the first step is to identify the parameter of interest which you wish to test, it could be the mean, it could be the variance of the population or the proportion of the population that you want to verify value. Now, you construct the null and alternative hypothesis as we said before. Then, based on a data experimental data about the system you collect and then you construct something called the test statistic. This is a function of the observations.

Now, this could be for example, if you are testing for the population mean you may use as the test statistic, the sample mean. If you are testing for the population variance you may use as test statistic the sample variance and so on, so forth. Now, you also have to derive the distribution of the test statistic under the null hypothesis, what it means

is if the null hypothesis is true what is the distribution of this test statistic that you are computed?

Now, based on this distribution you choose a threshold or the test criterion and now you compare the computed test statistic against this criterion and decide whether to reject the null hypothesis or not reject the null hypothesis. This is the overall procedure. We will show through examples how these procedure is carried out for different cases.

(Refer Slide Time: 05:46)

GyanData Private Limited

Hypotheses Testing Procedure

- No hypotheses test is perfect. There are inherent errors since it is based on observations which are random
- The performance of a hypotheses test depends on
 - Extent of variability in data
 - Number of observations (Sample size)
 - Test statistic (function of observations)
 - Test criterion (threshold)

40

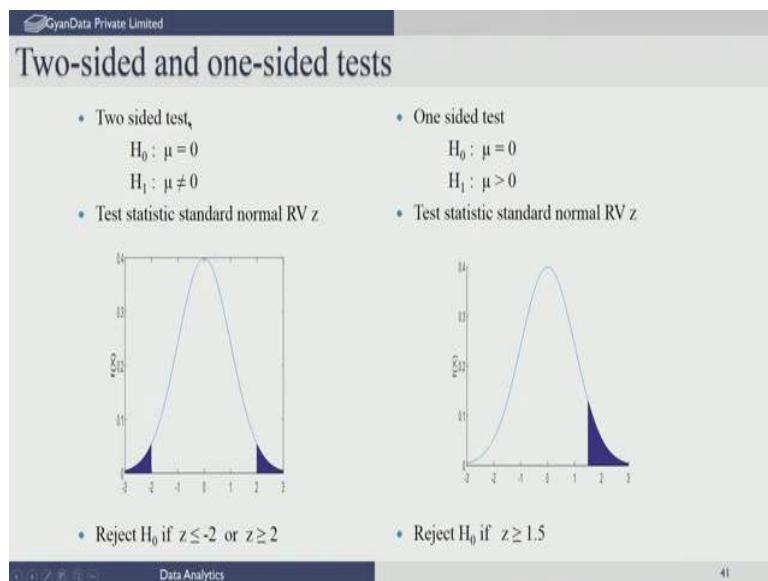
Now, you have to understand that the hypothesis testing procedure is an imperfect decision making process. Why? Because you are basing the test on a sample set of observations not on the entire population. So, there are inherent observe errors in the hypothesis testing because it is based on observations which are themselves stochastic in nature. Now, therefore, the performance of the hypothesis test, how well this decision making perform, depends on how much variability is there in the data. This you may not be able to do much about this they might be enough sufficient variability in the data which prevents you from making a good decision.

You can also alter the performance of the test by choosing the number of

observations, experimental observations, you want to make which is called the sample size. Now, the test statistic as we said is a function of the observations there are different sometimes you might have different choices of functions, and some test statistic may actually perform better than others. That depends on the theoretical foundations of statistical hypothesis testing, we will not touch upon this. We have control over the number of observations. We will take a look at how we can alter the performance based on sample size.

Finally, you also choose a threshold against which you are comparing the test statistic and therefore, you can alter the performance based on the criterion that you select, and we will see how this test criterion affects the performance of the hypothesis test.

(Refer Slide Time: 07:17)



Now, there are two types of hypothesis tests what we call the two sided test and the one sided test and I am giving a simple illustration to tell you what a two sided test means. Suppose you are testing for the population parameter, population mean μ , and you want to test the test whether this population parameter = 0 or $\neq 0$. So, the null hypothesis is the population parameter is 0, and the alternative is then a population mean is $\neq 0$. So, you observe some set of observations from this particular population we have a sample and you have computed let us say a sample statistic and let us also assume that sample statistic happens to be a standard normal variable z. We will show you how to construct a test statistic that finally, has this kind of a distribution for testing the mean.

But let us for the time being take it that we have a test statistic based on the observations we have made and this test statistic that we have computed is the standard normal test statistic z . Now, we know that this z , because it is a standard normal distribution, will have some shape like this and about 95 per-cent of the time the data the statistic will have a value between around - 2 to + 2. So, for a two sided test, we will say if the test statistic happens to be very large we will reject it or if it is very small we will reject it. Why? because if it is very large it means it does not come from a distribution with mean 0. So, in this particular case what we mean by very large we can choose a threshold let us say - 2, I am sorry + 2 and if the statistic is greater than 2 we reject the null hypothesis or if its small what do we mean by small if it is less than - 2 we reject the null hypothesis.

So, in this case we reject the null hypothesis whether the test statistic is less than a particular value the threshold value, in this case I have chosen the threshold as - 2, or if the statistic is greater than the threshold value, which is + 2. So, there are 2 thresholds the upper threshold which is 2 and the lower threshold which is - 2 because it is a two sided test. We want to reject the null hypothesis if μ is less than 0 or μ is greater than 0. How we choose these thresholds and what are the implications we will see later. But realize that if it is a two sided test you basically have a lower criterion threshold and the upper threshold which you select from the appropriate distribution.

Now, suppose you have the same thing, but you are only interested in testing whether the mean is 0 or greater than 0. So, in this case the null hypothesis is $\mu = 0$ and the alternative is μ less than 0 or greater than 0. So, notice that $\mu = 0$ implies that you are not interested in the case when μ is less than 0, you are you are not going to reject the null hypothesis if μ is less than 0 you are going to reject the null hypothesis only μ is greater than 0. That is why you have written the alternative like this. This is called the one sided test.

In this case let us assume that you again have computed a test statistic based on the observations and that is a standard normal test , then we only have an upper threshold, because we want to reject the null hypothesis only if the mean is greater than 0. So, similarly if the statistic is greater than a threshold then we reject the null hypothesis. So, we have a upper threshold, in this case I have chosen 1.5 and reject the null hypothesis if the statistic happens to be greater than 1.5. If it has a low value we are not bothered because we are not bothered when μ is less than 0.

So, although technically we should call the null hypothesis μ less than or $= 0$ it is usually stay in equality as $\mu = 0$ with indifference. Essentially the alternative tells you whether we are indifferent to μ less than 0 or not. So, this is called a one sided test. So, depending on the

type of test whether its two sided or one sided you choose thresholds and then compare the test statistics against those thresholds.

(Refer Slide Time: 11:41)

The slide is titled "Errors in Hypotheses Testing" and is from GyanData Private Limited. It includes a truth table and two bullet points:

- Two Types of errors (Type I and Type II)

Decision →	H_0 is not rejected	H_0 is rejected
Truth ↓		
H_0 is true	Correct Decision $Pr = 1 - \alpha$	Type I error $Pr = \alpha$
H_1 is true	Type II error $Pr = \beta$	Correct Decision $Pr = 1 - \beta$

- Typically the Type I error probability α (also called as level of significance of the test) is controlled by choosing the criterion from the distribution of the test statistic under the null hypothesis

Now, when you do such a test you commit two types of errors. So, essentially let us look at this truth table. Suppose the null hypothesis is actually true and you have made a decision to not reject the null hypothesis which means you have made the correct decision. So, this will not happen all the time because your sample is random it is possible that even if you are not high passes is true, you may conclude that the null hypothesis you may decide to reject the null hypothesis in which you commit a type I error what we call a type I error or a false alarm.

So, when the null hypothesis is true and then you reject the null hypothesis based on the sample data and your statistic and the threshold d of selection. So, you have selected, we call this a type I error or false alarm and the probability of that is known as α and we call it a type I error probability.

So, you will have it is not that your decision is perfect you will always commit some type I error α depending on the threshold that you have selected and the statistic you have computed. Similarly let us assume that the truth is that the alternative is correct. So, in this case it may turn out that from your sample setup data you do not reject the null hypothesis, in which case you commit what we call a type II error and this type II error also has a probability which is denoted by β .

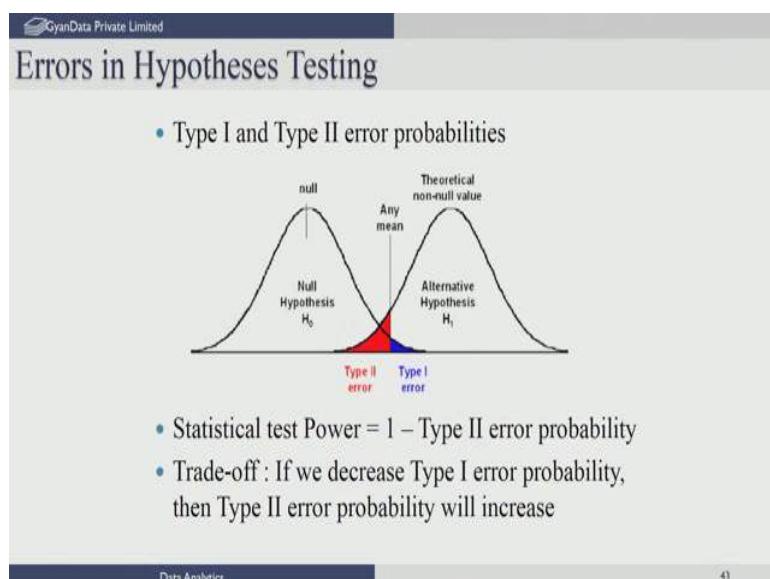
On the other hand if the alternative hypothesis is true and do you do reject the null hypothesis in favor of the alternative then you have

made a correct decision and that correct decision probability is known as power of the statistical test and is denoted by $1 - \beta$. Remember the only one of two decisions you have always going to make, you are either going to reject the null hypothesis or you are not going to reject it. So, the total property always be 1 and the probability of type II error if its β then the probability or statistical power is $1 - \beta$. So, there are two types of errors that you commit what to call the type I error probability α , and the type II error probability β .

So, the type I error probability α is also known as the level of significance of the test and this is typically what you control. You do not control the type II error probability. When you construct this hypothesis test to construct the test statistic and choose a threshold you basically try to control this type I error probability. The type II error probability results as a consequence of this. So, we will choose the criteria from the distribution of the test statistic under the null hypothesis because the null hypothesis is very precisely stated. If you go back and look at it we are able to state the null hypothesis precisely, the parameter value is 0.

Whereas, the alternative hypothesis we do not know what the parameter value we entertain a huge set $\mu_0 = 0$ involves everything other than 0. So, the parameter set is undefined very not clearly defined at least exact value is not clearly defined in the alternative. Whereas, for the null we are clearly defined in both cases. Therefore, it is possible to construct the distribution of the test statistic under the null hypothesis and that is the reason we only end up controlling the type I error probability and not the type II directly by choosing the test statistic.

(Refer Slide Time: 15:13)



Now, let us see how we actually control the type I error probability by choosing the appropriate test statistic value, test criterion value. So, let us look at the qualitative comparison of the type and type I error and type II error.

Let us assume that under the null hypothesis you have this distribution which is known as distribution of the test statistic under H_0 . Now, depending on, remember that even if H_0 is true, the statistic can have a value between $-\infty$ to $+\infty$. On the other hand you are going to choose some threshold in I have indicated this by this vertical line you are going to choose some threshold because you cannot say that you will accept you will not reject the null hypothesis whether the value is minor any value between $-\infty$ and $+\infty$. In which case you will never reject a null hypothesis whatever be the evidence you get that is not fair.

You decide to null reject the null hypothesis let us say in this case a one sided test and you have decided to reject it, if the statistic exceeds this threshold value. In which case notice that when the null hypothesis is true this test statistic can have a value greater than this threshold. And the probability that can have a value greater than the threshold is indicated by this blue area. So, this is your type I error probability that you have committed . If the null hypothesis is true your test statistic can exceed this threshold in which case this is the area the probability that that the test statistic can exceed this threshold and therefore, this is your type I error probability.

Now, if you move the threshold to the right obviously, you can reduce your type I error probability, but there is a price to be paid. Let us look at what is the price. Let us look at what happens to a type II error probability. So, let us assume that the actual distribution of the test statistic under H_1 under the alternative happens to be this kind of a distribution. Of course, this depends on what value the parameter is going to take.

Suppose we consider this distribution as the distribution of the test statistic if the alternative is true, then what happens is there is a probability that the test statistic will take a value less than this threshold and that is given by the probability marked in red, which means that even if the alternative is true you will not reject the null hypothesis because your test statistic is falling to the left of this threshold. And the probability that the statistic will be left of the threshold is given by the red area and that is going to be of type II error probability. And the power is just $1 - \beta$ which is the area to the right of this under this distribution, under H_1 . Now, notice that as you move the threshold to the right, the blue area shrinks which means your type I error shrinks, but your red area increases which means your type II error increases. So, there is a trade off.

If you try to make your test perfect in the sense of no type I error then you will come commit a type II error of one. Which means you are be a very insensitive test you will never be able to find whether your mean is different from 0 for example. So, the more less type I error you are willing to entertain the less sensitive your test will be. So, that is always a trade off you cannot help it. So, that there is a trade off if we decrease type I error probability then type II error probability will increase there is no choice and you have to accept this trade off.

So, you decide that I am willing to accept a type I error probability of 1 per-cent or 5 percent or 10 percent, and accordingly your test will be less or more sensitive. This is the way, the threshold selection is based on how much of false alarm probability or type I error probability you are willing to accept and that is the trade off you should accept and accordingly your test will be less or more sensitive.

(Refer Slide Time: 19:16)

GyanData Private Limited

Test for Mean : Solid Propellant example

For a given application the burning rate of a solid propellant should be 50 cm/s.

- 25 samples of the solid propellant are taken and their burning rate noted. The average burning rate is computed to be 51.3 cm/s. The standard deviation in the burning rate is known to be 2 cm/s
- Null hypothesis : $\mu = 50$ cm/s
- Alternative hypothesis : $\mu \neq 50$ cm/s (lower or higher burning rate propellants are both unsatisfactory) – Two sided test
- Test statistic $z = \frac{\bar{x}-50}{2/\sqrt{25}} \sim \mathcal{N}(0,1); z = 3.25$
- Critical value for $\alpha = 0.05$ is ± 1.96
- Decision: Reject null hypothesis

Data Analytics

Now, let us look at some examples for hypothesis testing. In this case we have looked at a manufacturer of a solid propellant and we want this solid propellant to burn at a certain rate and this burning rate of the solid propellant is specified to be let us say 50 cm per second. If it burns at a higher rate then we will not be able to come control the rocket, if it burns at the slower rate then the rocket may not even take off . So, we are going to check whether the propellant we are made, which is based on mixing a lot of different chemicals, whether it will have a burning rate of 50 centimeter per second.

Now, what we have done in the from the mixing bowl where we are making the solid propellant, we have taken 25 samples from different locations in the mixing bowl and each of these samples we test in the

lab and find that what their burning rate is. And it will be some value maybe it is 48, 49, 51 whatever. We compute the average of these 25 samples.

Notice, that the population parameter mean that we are testing is $\mu = 50$ and we are going to use the sample mean. We have taken 25 samples we compute them average or the sample mean of the burning rates of these 25 samples and we are going to make a judgment about the entire batch that we are making in the mixture remember. So, the population in this case is the entire batch of product that you are making and you are taking only a few samples and based on these samples you are making a judgment about the entire batch. So, the population parameter mean μ is what is what you are interested in you do not know what this value will be, you are going to ask this question whether that mean is going to be 50.

The sample mean that you have computed based on these 25 samples happens to be 51.3 centimeter per second, the burning rate average value. We have also computed the standard deviation based on these 25 samples we can compute the standard deviation of the sample and we find that the standard deviation is two centimeter per second. So, different samples have different burning rates and we find that the variability or the standard deviation in the burning rate of the samples is 2 centimeter per second.

Now, based on this data that we are collected the sample mean and the sample standard deviation we want to ask the question whether the population mean happens to be 50 centimeter per second. I am sorry let us say that the sample standard deviation is already known to you given to you that it will be 20, 2 cm per second it is not based on the sample its already known to you. Let us take that as a case which is the simplest case.

So, here we have actually said the null hypothesis population as having a burning rate of 50 centimeter per second and the alternative is $\mu_0 = 50$. Notice, it is a two sided test because we want to reject this batch if the burning rate is less than 50 centimeter per second it is not useful to us or if its greater than 50 centimeter per second that is why we have taken the alternative to be not $= 50$ centimeter per second. Now, this is a two sided test. Now, how do we construct the tests? I already talked to you that the sample mean has a normal distribution with population mean as the parameter the expected value and the standard deviation of the population divided by square root of n.

You can refer to the previous lecture to find out that the distribution of \bar{x} the sample mean, is the same as the population mean and has a standard deviation which is one standard deviation of the population divided by the square root of the number of samples you have taken. So, we can now do what is called standardization which is subtracting

the mean of the sample mean which is 50, the expected value ,divided by the standard deviation of the sample mean, which is 2 by root 25, and we get what is called the standardized value and this standardized value z will have a standard normal distribution with 0 mean and unit variance.

Now, notice that this will be the distribution if the population mean happens to be 50; that means, under H_0 under the null hypothesis this test statistic will have a standard normal distribution. Now, we know that 95 percent of the of a standard normal variable will lie between + or - 1.96 standard deviation approximately. More particularly the standard normal variable will lie between + or - 1.96, 95 percent of the time. So, if we are willing to tolerate a type I error probability of 0.5 percent let us say, which means what you are saying is, the area that you have on the left + the area that you have on the right of the upper threshold = 5 percent. If you are willing to tolerate that type I error probability then you can choose your criterion or your I am sorry your threshold as 1.96 and - 1.96 and that is exactly what is stated here.

The threshold value is + or - 1.96 and if your z statistic happens to be greater than 1.96 you reject it if, if it is less than one point - 1.96 you reject the null hypothesis in this particular case. If you substitute the value of the \bar{x} that we have obtained which is 51.3 and we have this standard deviation which you already know and we compute this we get a value of 3.25 which is greater than 1.96 and therefore, we reject the null hypothesis - this batch does not satisfy our needs. So, this is the way of testing and this is an example of testing for the mean given the standard deviation of the population.

In case your standard deviation of the population is not known, then you use the sample standard deviation and this test statistic we can show is a t statistic and therefore, you can derive your thresholds from your t distribution rather than the z distribution.

(Refer Slide Time: 25:41)

GyanData Private Limited

Test for Differences in Means : Training example

Two groups of teachers of similar capabilities are trained by two methods A and B. Is Method B more effective than Method A?

- 10 teachers in each group. Average scores and standard deviation of scores after training are. Group 1: $\bar{x}_1 = 70, s_1 = 3.3665$ Group 2: $\bar{x}_1 = 74, s_1 = 5.3955$
- Null hypothesis : $\mu_1 - \mu_2 = 0$
- Alternative hypothesis : $\mu_1 - \mu_2 < 0$ – one sided test
- Test statistic (assuming unknown but equal variances for two groups)
$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_p^2}{N_1} + \frac{s_p^2}{N_2}}} \sim t_{N_1+N_2-2}; S_p = \sqrt{\frac{(N_1-1)s_1^2 + (N_2-1)s_2^2}{N_1+N_2-2}} \quad t = -1.989$$
- Critical value for $\alpha = 0.05$ is -1.73
- Decision: Reject null hypothesis (Method B is better)

Data Analytics

Let us look at another example where we are testing for difference in means. Here we have two groups of teachers. This is a social example. So, we have two groups of teachers of let us say similar capabilities. There are 10 teachers in each group and we have trained them by two different methods A and B teaching methodology.

Now, we want to know whether method B is more effective than method A. Now, let us actually we assumed that we have 10 teachers in each group you have given training for one group of teachers using method A and another group of teachers using method B. And then we test them for the effectiveness and we find this course, we find that the average course of group 1 happens to be 70 with a standard deviation in the marks is 3.37, and group 2 has a sample mean of 74 with a standard deviation as 5.4 approximately.

So, we want to know whether method A and method B are both equally effective or method B is more effective than method A. This is the question that we want to do. So, we have chosen the null hypothesis that method A mean and method B mean both are equal which means their difference in the mean should be equal to 0 that is your null hypothesis. We will reject this in favor of the alternative which says the average the effectiveness of method A is less than the effectiveness of method B which means the mean of method A is less than the mean of method B , which means the difference should be less than 0. Notice that it is a one sided test because we are not interested in looking at whether method A is more effective than method B, we are only asking the question is method B more effective than method A and we are setting up our hypotheses accordingly. The difference in means is equal to 0 happens to be the null hypothesis, the difference in

means 1 - 2 is less than 0 is the alternative we choose, in case we find enough evidence for rejecting the null hypothesis.

Now, we will assume that the standard deviation of these two groups is same, although we do not know the standard deviations. So, we will take make the assumption that the standard deviation of marks obtained by group A population and the standard deviation of the group B population of teachers are equal, but unknown. But, however, we have the sample standard deviations from these two groups from which we can estimate the standard deviation of this entire group of teachers.

Notice that all the teachers I can group them because they have the same variability, there is no difference in the variability of group A and group B. So, we can pool their variances and we can obtain a pooled variance by just taking the sum square deviation of all with their respective means and then obtaining a pooled variance. This is a way by which you obtain the pooled variance for two groups, and once you obtain an estimate of the standard deviation of the group of teachers then you can take the difference in the sample means because remember you are testing the difference in means, so you can take as the test statistic in the sample means ($\bar{x}_1 - \bar{x}_2$) divided by the variance standard deviation of these means which is what this is all about. So, S_p represents the standard deviation of the difference in the means, remember assuming that we have they are both the groups have the same variances.

So, now, we can show that this particular statistic we have computed is t statistic because σ is also estimated from the data. So, we can now compare this with this t distribution, the number of degrees of the t distribution happens to be $N_1 + N_2 - 2$. Notice that that depends on the denominator degrees of freedom which is essentially total number of observations - 2 parameters which are mean parameters that you have used up to estimate the means. So, the remaining degrees of freedom is this and that is how this number of number of degrees of freedom comes about.

In fact, for large enough N_1 and N_2 , if you take large in a frame, you can actually approximate this with a standard normal variable , but in this case let us let us do a precise job. We will choose the test statistic test criteria from the t distribution with this many degrees of freedom $10 + 10 - 2$ which is 18 degrees of freedom and we find that the one sided confidence interval. Remember this is one sided if we are willing to tolerate a type I error probability of 5 percent then there is only a lower threshold less than 0 notice. So, - 1.73 the probability that a t distribution with 18 degrees of freedom is greater than this - 1.73 will be 5, 5 percent.

So, if we are willing to tolerate the type I error probability of 5 percent then we can choose the threshold value as - 1.73 drawn from the t distribution with eighteen degrees of freedom and compare it with the statistic. The test statistic itself we compute by plugging in the value for \bar{x}_1 , \bar{x}_2 and so on so forth we get - 1.989, since this is less than - 1.73 we reject this null hypothesis in favor of this alternative, which means method B is more effective than method A that is a conclusion.

(Refer Slide Time: 31:27)

The variability in yields from two different processes are to be compared to decide whether they are identical or not

- 50 samples for each process taken. Yield variances are found to be $s_1^2 = 2.05$ and $s_2^2 = 7.64$
- Null hypothesis : $\frac{\sigma_1^2}{\sigma_2^2} = 1$
- Alternative hypothesis : $\frac{\sigma_1^2}{\sigma_2^2} \neq 1$ – two sided test
- Test statistic (assuming unknown but equal variances for two groups)

$$f = \frac{s_1^2}{s_2^2} \sim F(N_1 - 1, N_2 - 1); f = 0.27$$
- Critical value for $\alpha = 0.025$ is 0.567 and $\alpha = 0.975$ is 1.762
- Decision: Reject null hypothesis (Process 2 has higher variability)

So, let us look at another example which is a test for difference in variances. Now, we have two processes which have different variability and we want to compare whether these two process have the same variability or not. So, we have taken 50 samples from each process and computed their variances and found that one has a variability of 2.05 and the other has a variability or a variance of 7.64. Now, what we want to compare is whether these two variances, population variances, of these two process are equal or not which means the ratio of the variances we want to check whether it = 1 or not.

The alternative hypothesis is that these variances are different, so we are asking whether the ratio of the variance is not equal to 1. It could be that σ_1^2 is less than σ_2^2 or σ_1^2 is greater than σ_2^2 both cases we want to reject the null hypothesis. So, this is a two sided test.

Again we can use the ratio of the sample variances as a test statistic and this ratio turns out to be an f distribution with degrees of freedom $N_1 - 1$ and $N_2 - 1$ where N_1 and N_2 represents a number of samples we have taken for each of the process. In this case we have equal number of samples we have taken, even if you are taken unequal samples we can appropriately choose the degrees of freedom for the f distribution

and compare. So, in this case if you plug in the values for S_1^2 , S_2^2 we have got a F value of 0.27, and if you go to the F distribution with $N_1 - 1$ which is 49 degrees of freedom and $N_2 - 1$ which is 49 degrees of freedom and ask 5 percent probability which we break it up as two, left of the threshold should be 2.5 percent and greater than the threshold is 2.5 percent.

So, we ask what is the threshold where lower threshold value for the F distribution and it turns out to be 0.567 that means 2.5 percent, there is a 2.5 percent probability that this F distribution is less than this value. Similarly under H_0 and similarly there is A_{2.5} percent probability that the F distribution is greater than this value. So, the lower threshold we choose as 0.567 upper threshold is 1.762.

The statistic we are computed happens to be less than this. So, we reject the null hypothesis in favor of the alternative and claim that that the two variances of the two process are not equal. Of course, although this implies that the process two has a higher variability, if you really wanted to test this and not worry about process one then we had have set it up differently, the hypothesis testing.

(Refer Slide Time: 34:24)

Type of test	Characteristic	Example	Application
Z-test	Sum of independent normal variables	Test for a mean or comparison between two group means (variance known)	Test coefficients of a regression model
t-test	Ratio of a standard normal variable and chi-square variables with p degrees of freedom	Test for a mean or comparison between two group means (variance unknown)	Test coefficients of a regression model
chi-square test (p degrees of freedom)	Sum of p independent standard normal variables	Test for variance	Test quality of regression model
F-test (p_1 and p_2 degrees of freedom)	Ratio of two chi-square variables	Test for comparing variances of two groups	Choose between regression models having different number of parameters

In summary I want to just go over some of the standard tests that we use in hypothesis testing and see where they are useful. So, the z-test or the standard normal variable statistic which has a standard normal variable, we call it a z-test. And this is typically used for testing of the mean or a comparison between two means when the variance of the population is known.

The application of such a test is usually found in testing whether the coefficient of a regression linear regression model is 0 or not. Now, the t test on the other hand is used also for a test of the mean or a comparison between means group means, but in this case the population variance is not known. So, we use the sample variance for normalization and that gives rise to a t test. Here also we whenever we test for the coefficients of the regression model under the assumption that the errors corrupting the variables, we do not have an idea the way a standard deviation of the errors that corrupt the observations are unknown, then we use the t test and of to test whether the coefficient of regression model = 0 or not.

The χ^2 square test is used for testing the variance of a sample and the variance of a sample, this test is used to whether a regression model is high is good or not, whether acceptable quality or not. The objective function of a regression model is the sum squared term and is similar to a variance, and you can use it to this χ^2 square test to test whether the integration model is acceptable or not.

The f test is used when for comparing variances of two groups. In the case of linear regression this is used to choose between two regression models having different number of parameter. Sometimes you might actually build a model by dropping a variable and you want to know whether the model you have built by reducing the number of variables is better than the 1, that by retaining all of them, in such a case this f test comes in very handy.

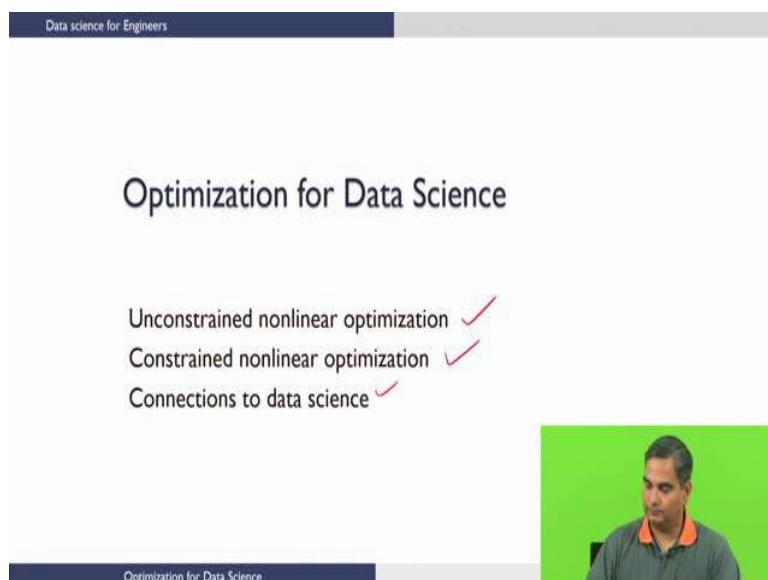
We have completed the introduction to probability and statistics. We will see you in the next lecture which we will talk about linear regression and the application of these concepts to linear regression.

Thank you.

Data science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 23
Optimization for Data Science

(Refer Slide Time: 00:11)

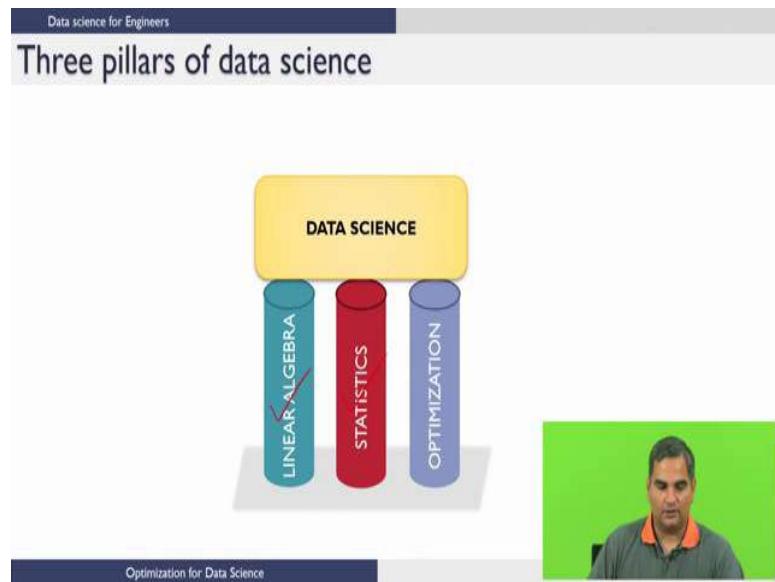


The slide has a dark blue header bar with the text "Data science for Engineers". The main title "Optimization for Data Science" is centered above a list of three topics: "Unconstrained nonlinear optimization" (with a red checkmark), "Constrained nonlinear optimization" (with a red checkmark), and "Connections to data science" (with a red checkmark). At the bottom left is a dark blue footer bar with the text "Optimization for Data Science". On the right side, there is a video player window showing a man from the chest up, wearing a dark polo shirt, against a green background.

In this series of lectures now, we will look at the use of optimization for data science. We will start with a general description of optimization problems and then we will point out the relevance of understanding this field of optimization from a data science perspective.

We will also introduce you very briefly to the various types of optimization problems that people solve. While all of these types of problems have some relevance from a data science perspective, we will focus on two types of optimization problems which are used quite a bit in data science. One is called the unconstrained non-linear optimization and the other one is constrained nonlinear optimization and as I mentioned before we will also describe the connections to data science.

(Refer Slide Time: 01:21)



I would really consider from a mathematical foundations viewpoint that the three pillars for data science that we really need to understand quite well are linear algebra which you already seen before. Following that you saw series of lectures on statistics and the third pillar really is optimization, which is used in pretty much all data science algorithms. And quite a bit of the optimization concepts for one to understand quite well you need a good fundamental understanding of linear algebra which is what we have tried to deliver through the series of lectures on linear algebra.

(Refer Slide Time: 02:04)

The diagram features a dark blue rounded rectangle containing the text: "An optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function."* Below this, a question is asked: What is optimization ? To the right is a video frame of a man speaking. The bottom left corner has the text 'Optimization for Data Science'.

So, we will start by asking what is optimization and Wikipedia defines optimization as a problem where you maximize or minimize a real function by systematically choosing input values from an allowed set and computing the value of the function, we will more clearly understand what each of these means in the next slide.

Now, when we talk about optimization we are always interested in finding the best solution. So, we will say that I have some functional form that I am interested in and I am trying to find the best solution for this functional form.

(Refer Slide Time: 02:30)

Data science for Engineers

What is optimization?

- ... the use of specific methods to determine the “best” solution to a problem
 - Find the best functional representation for data
 - Find the best hyperplane to classify data

$f(x)$
 $x = \text{decision variables}$

$y = A_0 + A_1 * x$

ϵ_i

Margin

Optimal

$\sum \epsilon_i$

Optimization for Data Science

Now, what does best mean? You could either say I am interested in minimizing this functional form or maximizing this functional form. So, this is the function for which we want to find the best solution. And how do I minimize or maximize this functional form? I have to do something to minimize or maximize and the variables that are in my control so that I can maximize or minimize this function or these variables x . so these variables x or call the decision variables.

And in the previous slide we talked about these being in an allowed set. What basically that means, is while I have the ability to choose values for x , so that this function f is either maximized or minimized, there would be some constraints on x which would force us to choose x in only certain regions or certain sets of values for this optimization problem. So, in that sense I have an objective which I am trying to maximize or minimize. I have decision variables which I can choose values for, that will either maximize or minimize the function. However, I might not have complete control over this x there might be some restrictions on x which are the constraints on x which I have to satisfy while I solve this optimization problem.

Now, why is it that we are interested in optimization in data science? So, we talked about two different types of problems, one is what is called a function approximation problem which is what you will see as regression later. So, in that case we were looking for solving for functions with minimum error remember that. Now, the minute I say minimum error, then I have the following minimum error, we said we have to define what this error is somehow, and the minute we say minimum error that basically means we are trying to find something which is the best we are trying to minimize something.

So, this part is already there finding a best for some function. And this error is something that we define. So, for example, if you remember back to our linear algebra lectures we said if there are many equations and they cannot be solved with a given set of variables then we said we could minimize this $\sigma_i = 1$ to $m e_i^2$.

So, this is the function now that we are trying to minimize. And there are the decision variables in that particular case we said the variable values are going to be the decision variable. So, this whole function if I call this as f , this is going to be a function of x - the values that the variables take. So, you already have a situation where you are trying to minimize a function and these are the decision variables. This is completely unconstrained or I have no restrictions on the values of x I choose, I can choose any value of x , I want as long as that value minimizes this or finds a best value for this function.

Now, remember the other case that we saw where we looked at much more variables than equations. In that case again we minimize the norm of the solution as the objective again there is a minimization there is an objective. However we said that optimization problem is constrained by the fact that the solution that I get should satisfy the equation. So, $ax = b$ is a constraint there in which case I am constraining of all the x 's that I can take I am constraining to those which would satisfy the equation $ax = b$. So, this idea of representation is used quite a bit in data science.

Another way to think about the same problem is the following, if I give you data for y and x and let us say you are trying to fit a function between y and x , so, you might say $y = a_0 + a_1 x$. Then what you have here is the following. So, I might give you several samples. So, I have $y_1 x_1, y_2 x_2, \dots, y_n x_n$. So, I have given you several samples and I have told you that this is the model that you need to fit.

So, if I put each of the sample points into this equation. So, I will get $y_1 = a_0 + a_1 x_1$ and all the way up to $y_n = a_0 + a_1 x_n$. Clearly you can take the view that there are n equations here, but only two variables. So, there are many more equations and variables. So, I cannot solve all of these equations together. So, what I am going to do is I am going to define an error function which is very similar to what we saw before $y_1 - a_0 - a_1 x_1$ all the way up to $y_n - a_0 - a_1 x_n$. And then I know that I have only two variables that I can identify which are a_0 and a_1 ; however, there are n equations. So, what I am going to do is I am going to minimize a sum of squared error is something that we talked about.

Now, this error is going to be a function of the two parameters a_0 and a_1 . So, these become the decision variables and this becomes a function. And if you have no constraints on what the values can take, what the values these variables can take, then you have an unconstrained optimization problem. This is the type of problem that you would solve in linear regression and in general this is also called as function approximation problem. So, this is one type of problem which is used quite a bit in data science because in many cases we are looking at functional relationships between variables. So, that is one reason why optimization becomes important.

Now, in terms of the other bullet point I have here which is find the best hyper plane to classify this data. This is also something that we had seen before, where we looked at data points and then we said for example, I could have lots of data here corresponding to one class this I described when we are talking about linear algebra and I could have lot of data points here corresponding to another class. Now, I want to find a hyperplane which separates this.

Now, you could ask the question as to which is the best hyperplane that separates this. So, you could say I could draw a hyperplane here or I could draw hyperplane here or I could draw a hyperplane here and so on. Now, which one should I choose and the minute I say which one should I choose. We know that these hyper planes are represented by an equation. Then we say which hyperplane do I choose, then basically it means I am saying which equation do I use, which basically means what are the parameters in the equation that I choose to use. So, I want to find the parameters parameter values that I should use in that equation. So, those become the decision variables the parameters that characterize these hyper planes become the decision variables.

And in this case the function that I am trying to optimize is that when I choose a hyperplane I should not miss classify any data. So, for example, I have to choose a hyperplane in such a way that all of this data is to one half space of the hyperplane and all of this data is to the other half space of the hyperplane. So, you see that again this classification problem becomes an optimization problem.

(Refer Slide Time: 11:41)

So, in summary we can say that almost all machine learning algorithms can be viewed as solutions to optimization problems and it is interesting that even in cases, where the original machine learning technique has a basis derived from other fields for example, from biology and so on one could still interpret all of these machine learning algorithms as some solution to an optimization problem. So, basic understanding of optimization will help us more deeply understand the working of machine learning algorithms, will help us rationalize the working.

So, if you get a result and you want to interpret it, if you had a very deep understanding of optimization you will be able to see why you got the result that you got. And at even higher level of understanding you might be able to develop new algorithms yourselves.

(Refer Slide Time: 12:42)

Data science for Engineers

Components of an optimization problem

- Objective function f
- We look at minimization problem
- Decision variables x
- Constraints

Optimization for Data Science

So, as we have described in quite detail till now, an optimization problem has three components the first component is an objective function f which we are trying to either maximize or minimize. In general we talk about minimization problems this is simply because if I have a maximization problem with f , I can convert it to a minimization problem with $-f$. So, in without loss of generality we can look at minimization problems. So, that is one component in an optimization problem.

The second component are the decision variables which we can choose to minimize the function. So, I write this as $f(x)$. So, this is a function and these are the decision variables and our goal is to

minimize. And the third component is the constraint which basically constrains this X to some set that will be defined as we go along. So, whenever you look at an optimization problem. So, you should look for these three components in an optimization problem. In cases where this is missing we call this as unconstrained optimization problems, in cases where this is there and we have to have the solution satisfy these constraints we call them as constrained optimization problems.

(Refer Slide Time: 14:08)

Types of optimization problems

- Depending on the type of objective function, constraints and decision variables
 - Linear programming problem ✓
 - Nonlinear programming problem ✓
 - Convex vs Non-convex
 - Integer programming problem (linear and nonlinear)
 - Mixed integer linear programming problem
 - Mixed integer nonlinear programming problem

Now, depending on the type of objective function, type of constraints and the type of decision variables, we will explain what each one of these are, there are different types of optimization problems that we could solve.

For example, if we have the following $f(x)$ subject to some constraints that we are going to impose and if it turns out that this X we use them as continuous variables. What do we mean by continuous variables? These are variables that can take values within a certain range. So, you could say if you have one variable you could say the variable could be between - 2 and 2 for example, or you could simply say it could be any number in the real line then these are condensed variables. What it basically means is within this range I can take any value there is no restriction on the value right X . So, these are continuous variables. So, if you have continuous variables like this, and if the functional form of this f is linear and all the constraints are also linear then I have a type of problem called linear programming problem.

So, in this case the variables are continuous, the objective is linear and the constraints are also linear. Now, if the variable remains continuous however, if either the objective function or the constraints

are non-linear functions, then we have what is called a nonlinear programming problem. So, a programming problem becomes non-linear if either the objective or the constraints become non-linear.

In general people used to think non-linear programming problems are much harder to solve than linear programming problems which is true in some cases, but really the difficulty in solving non-linear programming problems is mainly related to this notion of convexity. So, whether a non-linear programming problem is convex or non convex is an important idea in identifying how difficult the problem is to solve.

So, this idea of convex and non convex very very briefly without too much detail we will see in the next few slides nonetheless I just wanted to point this out here and also wanted to describe the second type of optimization problem that is of interest which is the nonlinear programming problem.

Till now, we have just been talking about the types of objective functions and constraints however, we have always assumed that the decision variables are continuous. In many cases we might want the decision variable not to be continuous, but to be integers. So, for example, I could have an optimization problem where I have f as a function of let us say two variables x_1 and x_2 and I could say minimize this. Now, I could say x_1 is not continuous, but x_1 has to take a value let us say from this integer set $\{0, 1, 2, 3\}$ so on, and x_2 maybe has to also take a value in this set. So, this is called a integer programming problem.

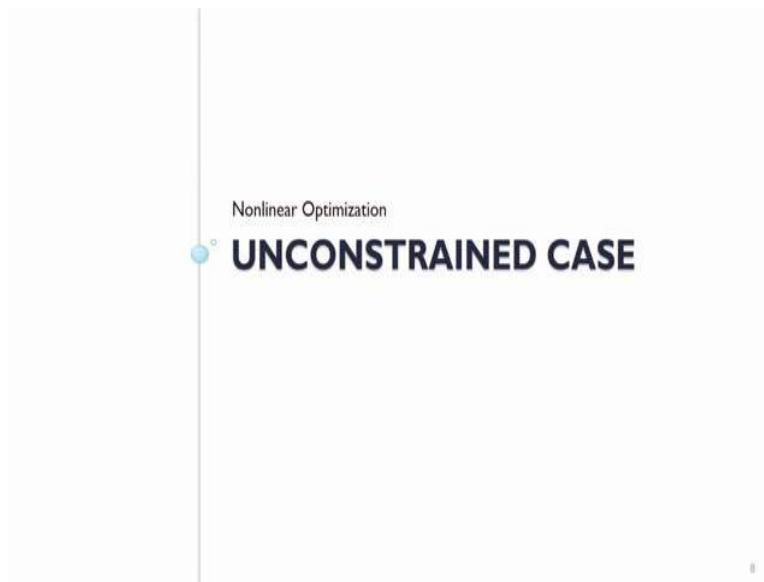
And you could have constraints also on x constraints on x_1 and x_2 could also be there. And if the objective function and constraints are linear then we call this linear integer programming problem, if either the objective or the constraints become non-linear we call them non-linear integer programming problems. One special class of these integer programming problems are binary where if x_1 could only take a value which is 0 or 1 and x_2 could take a value only 0 or 1 we call this as binary integer programming problems.

Now, when you combine variables which are both continuous and integer. So, for example, in this case when I have $f(x_1, x_2)$ let us say x_1 has to take a value 0 1 2 3 whereas, x_2 is continuous it can take any value let us say within a range then we have what are called mixed programming problems and if both the constraints, and the objective are linear then we have mixed integer linear programming problem and if either the constraints or the objective become non-linear then we have mixed integer non-linear programming problem. So, these are the various types of problems that are of interest.

Now, these types of problems have been solved and are of large interest in

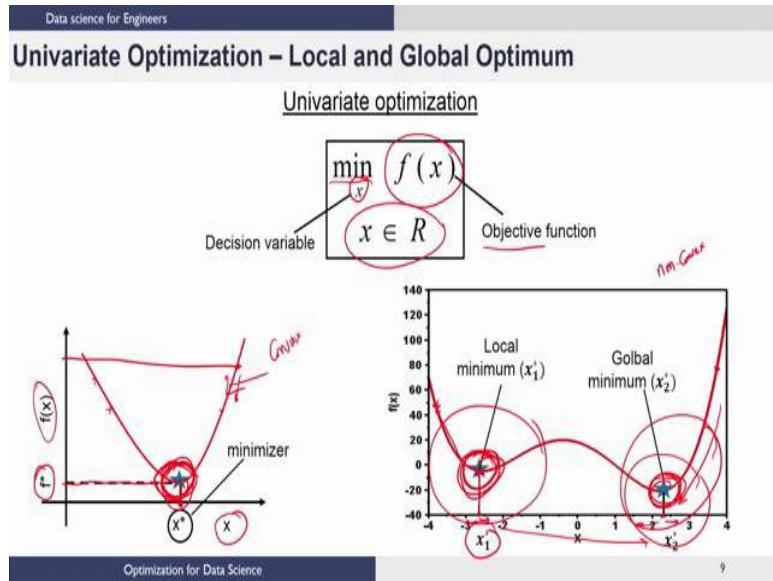
almost all engineering disciplines. So, we typically solve these problems in for example, in chemical engineering we solve these types of problems routinely for optimizing. Let us say refinery operations or designing optimal equipment and so on, and similarly in all engineering disciplines these optimization problems are used quite heavily. From these lectures viewpoint what we want to point out is to show how we can understand some of these optimization problems and how they are useful in the field of data science.

(Refer Slide Time: 20:48)



So, I am going to start with the simple case of a non-linear optimization problem unconstrained case that is there are no constraints.

(Refer Slide Time: 20:57)



Let us start with a very simple unconstrained optimization problem called an univariate optimization problem and in this slide I am going to explain this univariate optimization problem and the ideas of local and global optimum. So, what do we mean by univariate? When we say it is a univariate optimization problem there is only one decision variable that we are trying to find a value for.

So, when you look at this optimization problem you typically write it in this form where you say I am going to minimize something, this function here, and this function is called the objective function. And the variable that you can use to minimize this function which is called the decision variable is written below like this here x and we also say x is continuous, that is it could take any value in the real number line. And since this is a univariate optimization problem x is a scalar variable and not a vector variable. And whenever we talk about univariate optimization problems, it is easy to visualize that in a 2-D picture like this. So, what we have here is in the x axis we have different values for the decision variable x and in the y axis we have the function value. And when you plot this you can quite easily notice that, this is the point at which this function right here attains its minimum value.

So, the point at which this function attains minimum value can be found by dropping a perpendicular onto the x axis. So, this is actual value of x at which this function takes a minimum value and the value that the function takes at its minimum point can be identified by dropping this perpendicular onto the y axis and this f^* is the best value this function could possibly take. So, functions of this type are called convex functions because there is only one minimum here. So, there is

no question of multiple minima to choose from. There is only one minimum here and that is given by this.

So, in this case we would say that this minimum is both a local minimum and also a global minimum. We say it is a local minimum because in the vicinity of this point this is the best solution that you can get. And if the solution that we get the best solution that we get in the vicinity of this point is also the best solution globally then we also call it the global minimum.

Now, contrast that with the picture that I have on the right hand side. Now, here I have a function and again it is a univariate optimization problem. So, on the x axis I have different values of the decision variable on y axis we plot the function. Now, you notice that there are two points where the function attains a minimum and you can see that when we say minimum we automatically actually only mean locally minimum because if you notice this point here in the vicinity of this point this function cannot take any better value from a minimization viewpoint. In other words if I am here and the function is taking this value if I move to the right the function value will increase which basically is not good for us because we are trying to find minimum value, and if I move to my left the function value will again increase which is not good because we are finding the minimum for this function.

What this basically says is the following. This says that in a local vicinity you can never find a point which is better than this. However, if you go far away then you will get to this point here which again from a local viewpoint is the best because if I go in this direction the function increases and if I go in this direction also the function increases, and in this particular example it also turns out that globally this is the best solution. So, while both are local minimum in the sense that in the vicinity they are the best this local minimum is also global minimum because if you take the whole region you still cannot beat this solution.

So, when you have a solution which is the lowest in the whole region then you call that as a global minimum. And these are types of functions which we call as non convex functions where there are multiple local optima and the job of an optimizer is to find out the best solution from the many optimum solutions that are possible.

Now, I just want to make a connection to data science here. Now, this problem of finding the global minimum has been a real issue in several data science algorithms. For example, in the 90s there was a lot of excitement and interest about neural networks and so on, and for a few years lot of research went into neural networks and in many cases it turned out that finding the globally optimum solution was very difficult and in many cases these neural networks trained to local

optima which is not good enough for the type of problems that were that being solved.

So, that became a real issue with the notion of neural networks and then in the recent years this problem has been revisited and now there are much better algorithms, and much better functional forms, and much better training strategies, so that you can achieve some notion of global optimality and that is reason why we have these algorithms make a comeback and be very useful.

So, this very simple concept of local and global optimization is a very important challenge in many data science algorithms and we will see those later. I just also want to point out why this becomes a challenge. This becomes a challenge because when you run a data science algorithm depending on where you start the algorithm you will get different solutions if the problems are non-convex.

So, in other words whenever you solve an optimization problem as we will see later, you will start with some initial point and try to keep improving your function value by changing the value of your decision variable. So, for example, if you started here for this problem and the function value is something like this you know that if you want to improve your function value that is it since you are minimizing you want to reduce your function value you have to keep going in this direction. And what will happen is ultimately you will get to this point and then say I cannot improve my objective function anymore. So, this is the best solution that is possible.

This is how most optimization algorithms work. An important thing to notice here is the respective of whether you start here or here or here or here you are likely to go here depending on your algorithm, you can go there quicker you can go the slower and so on nonetheless whatever is your initialization you are likely to get to the same solution. So, in other words when this optimization algorithm is the backbone of your data science algorithm every time you run the data science algorithm you will get the same solution.

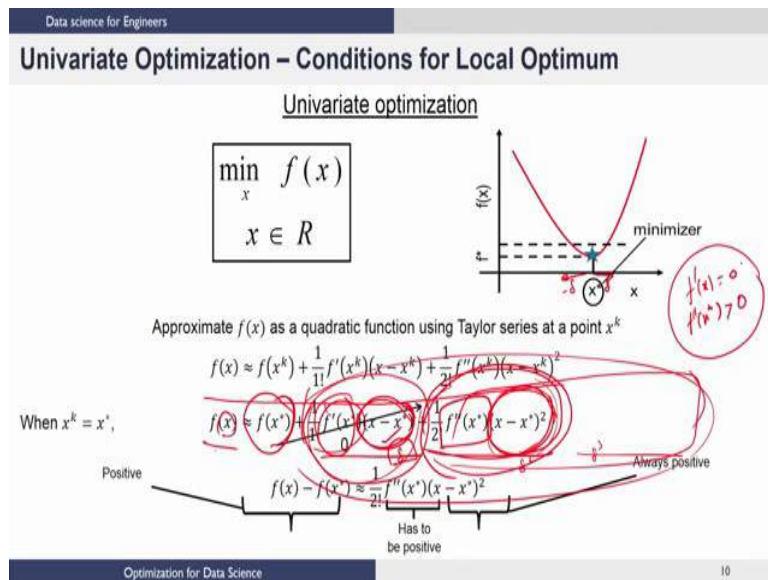
However, notice this picture right here for example, if I started here I want a better my function. So, I will keep improving it and when I come here there is no way to improve it any further. So, I might call this as my best solution and then the data science algorithm will converge. However, if I start here then I would more likely end up here and then I will say this is the best I get and I will stop my data science algorithm.

Now, notice what happens in this case if your data science algorithm is trying to find a value for the decision variable, when you run this once with this initialization you might get this as a solution to

your problem, and when you run with this initialization you might get this as a solution to this problem. In other words the algorithm will not give you the same result consistently and more importantly if it is very difficult to find this most of the time your algorithm will give you result which is local minimum, in other words you could do much better, but you are not able to find the solution that does much better.

So, this is an important concept and that you want to understand later when we show you data science algorithms and show you several runs of the same data science algorithm you get several results you might wonder why that is happening and that is due to this problem of initialization.

(Refer Slide Time: 30:48)



Now, let us look at conditions for value to be a minimizer. So, we take the same problem minimize $f(x)$, x element of R . Now, these are conditions that you have seen many times before I am just going to quickly describe how we derive these conditions.

So, if I take $f(x)$ and then let us say I am at a particular point x_k , what I can do is I can do a taylor series approximation of this function which we would have seen before in high school and so on. So, let us say this x^* is the minimum point and let us see what happens to this Taylor series approximation around this point. So, I am going to say this function $f(x)$ can be approximately written as $f(x^*) + \dots$. Now, if you notice this expression right here this is a number because this is an x^* that I know. So, I simply evaluate f at that x^* . So, this is a number, so this is not a function of this x . However, the second term and third term and so on we will all be functions of x . In other words if I change x these are the terms that will change this will remain the same.

Now, you could see that if you look at this term this is $x - x^*$ if you look at this term this is $x - x^*$ square and so on. In this univariate case let us call this as δ . So, if I go a δ distance from this $- \delta$ here, let us call this $x - x^* \delta$. So, if I go in the positive direction I will have a δ , I will have a δ^2 , I will have δ^3 and so on. Now, this is a fixed number let us look at this sum of these terms. Now, if you keep reducing δ to smaller and smaller values this is what we explained when we said we are looking at it locally. So, at some point what will happen is δ will become so small that none of these terms will matter the sign of the whole sum will be only depending on this term here. So, if this term is positive this whole sum will be positive and if this term is negative the whole sum will be negative.

Now, you notice this and then if you look at this, if let us say this sign is positive for positive δ then, unfortunately when I go in the negative direction it will become negative because this is again a fixed number if this is positive for δ for $- \delta$ this will become negative. That basically means that x^* cannot be a minimizer because I can further reduce this function by going to the left.

Now, when δ is positive if this function turns out to be negative then I can go in the to the right and then minimize my function again. So, if this term is not 0 then for sure I will have one direction in which I can go and find a value better than $f(x^*)$ locally, which would invalidate our argument that x^* is a minimizer. So, basically the only way out for this x^* to be minimizer is for this term to be 0 irrespective of x that basically means that $f'(x)$ has to be 0. So, that is the first condition that we usually get $f'(x)$ is 0 and once this is 0 then the Taylor series expansion basically becomes $f(x)$ is $f(x^*) +$ the second term third term and so on. By using the same argument when δ becomes smaller and smaller and smaller this term is the only term that will determine the sign of the sum.

However, notice something very interesting and different here. When we looked at this term this was $x - x^*$, when we look at this term now it is $(x - x^*)^2$. Now, this term, the sign of this term, is dictated only by this quantity here because this is a square and it will always be positive. So, if this $f(x)$ has to be minimized at x^* then basically this number $f''(x^*)$ has to be greater than 0 because if this is greater than 0 irrespective of whether you are going to the left or the right this is always positive. So, this will always be a positive contribution; that means, $f(x)$ will always be greater than $f(x^*)$ in the local region which should make x^* a minimizer. So, that is the important idea in varied optimization and that is the reason why you get these two conditions.

(Refer Slide Time: 35:54)

Data science for Engineers

Univariate Optimization – Summary

Univariate optimization

$$\min_x f(x)$$

$$x \in R$$

Necessary and sufficient conditions for x^* to be the minimizer of the function $f(x)$

First order necessary condition: $f'(x^*) = 0$ ✓

Second order sufficiency condition: $f''(x^*) > 0$ ✓

Optimization for Data Science

So, in summary the first order necessary condition as we call it is that the first derivative with respect to x when evaluated at x^* has to go to 0. And the second order sufficiency condition as we call it is that then I evaluate the second derivative with respect to x and then evaluated at x^* it has to be greater than 0.

(Refer Slide Time: 36:20)

Data science for Engineers

Univariate Optimization – Numerical Example

$$\min_x f(x)$$

$$f(x) = 3x^4 - 4x^3 - 12x^2 + 3$$

First order condition	Second order condition
$f'(x) = 12x^3 - 12x^2 - 24x = 0$ $= 12x(x^2 - x - 2x) = 0$ $= 12x(x+1)(x-2) = 0$ $x = 0, x = -1, x = 2$	$f''(x) = 36x^2 - 24x - 24$ $f''(x) _{x=0} = -24$ $f''(x) _{x=-1} = 36 > 0$ $f''(x) _{x=2} = 72 > 0$
$f(-1) = -2$ $x^* = -1$, is a local minimizer of $f(x)$	$f(2) = -29$ $x^* = 2$, is a global minimizer of $f(x)$

Optimization for Data Science

Let us quickly through a see a numerical example to bring all of this ideas together. So, let us take a function $f(x)$ which is of the form $3x^4 - 4x^3 - 12x^2 + 3$. Let us first do the first derivative and set it to 0, when

we do the first derivative and set it to 0 we get 3 solutions $x = 0$, $x = -1$ and 2 .

Now, we want to know which one of this is a minimizer and which one is a local minimizer global minimizer and so on. To do that we look at the second order conditions and then we get $f''(x)$ the second derivative and then we first evaluate it at $x = 0$. In this case this number turns out to be negative which means that x is a maximum point, not a minimum point. Our interest is in minimization and when we look at this f double prime at -1 and 2 the only thing we can look for is whether this number is positive or not. The actual numbers do not matter.

So, in this case this is 36 this is 72 in both cases this is greater than 0 . So, points $x = -1$ and 2 both are minimum points for this function because both of them satisfy the two conditions $f'(x^*) = 0$ and f double prime x^* is greater than 0 . Now, it is interesting that at this point we cannot say anything more about these two points these numbers do not help we just look whether they are positive or not and of these two points clearly one of them is a local minimum another one is a global minimum. So, the only way to figure out which point is a local minimum which is a global minimum is to actually substitute this into the function and then see what values you get. So, when you substitute -1 into the function you get -2 and when you substitute 2 into the function you get -29 . Since we are interested in minimizing the function -29 is much better than -2 . So, that basically means 2 is a global minimum of this function and -1 is a local minimizer for $f(x)$.

So, in this lecture we looked at simple univariate unconstrained optimization. And we also looked at why optimization is very important from a data science viewpoint. We will pick up on some of these ideas and then talk about multivariate unconstrained optimization and constrained optimization in the lectures to follow.

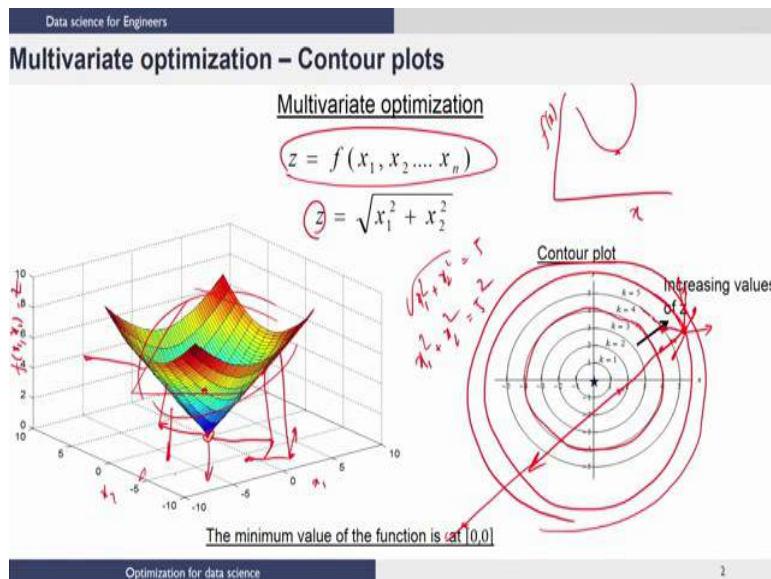
Thank you. I will see you again in the next lecture.

Data Science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 24
Nonlinear Optimization Unconstrained Multivariate Optimization

In the previous lecture we described unconstrained non-linear optimization in the univariate case or when there was only one decision variable. In this lecture we are going to see how this is extended to cases where there are multiple variables that act as decision variables in the optimization problem.

(Refer Slide Time: 00:36)



So, when you look at these types of problems, a general function z could be some non-linear function of decision variables x_1 to x_n . So, there are n variables that we could manipulate or choose to optimize this function z . A simple demonstration of this in a two dimensional case is root of $(x_1^2 + x_2^2)$.

Notice that we could explain univariate optimization using pictures in two dimensions that is because in the x direction we had the decision variable value and in the y direction we had the value of the function. So, you could see something like this and then say this is the minimum and so on. However, when you just extend this problem to two dimensions then you have to have 3-dimensional plots and in

dimensions higher than 2, if the decision variables are more than 2 then it is difficult to visualize. So, what we are going to do is we are going to explain some of the main ideas in multivariate optimization through pictures such as the one that I have shown on the left side of the slide and as I mentioned before since even for cases where there are two decision variables we need to go to 3-dimensions and that is simply because if this is x_1 and this is x_2 or this is x_1 and this is x_2 , I need a third dimension to describe the value of the function $f(x_1, x_2) = 0$. So, the objective function value becomes the third axis.

So, let us look at how we think about this unconstrained optimization when there are multiple variables. Take this picture for example, if you look at this picture right here on the left hand side of the slide you will notice that the minimum point is somewhere around here, which in this case happens to be 0 0 this is touching the x, y, x_1, x_2 plane and that is a solution minimum point is 0.

Nonetheless if you start moving in the x_1, x_2 plane then when you compute the objective function at different points. Let us say I compute the objective function at this point then this is going to be outside the plane and this is a value of the objective function at this point. And if I compute the objective function at this point it is going to be outside the plane this is going to be the value of the objective function at this point and so on and if I go this direction I might come here and so on.

So, what we are going to do is we are going to be in the space of decision variables and we are going to try and find an optimum solution because those are the values that we are actually choosing. So, for example, if let us say I have a point here on the space of the decision variables then the corresponding objective function value is this and clearly we know that that is not the minimum point.

So, what we need to do is we have to figure out some how to get here. Notice that the point at which the decision variables take values such that the function is a minimum is also in the decision variable space. So, essentially when we keep changing the values for the decision variables we are basically moving in this plane; however, while we are moving this plane we are looking at the values in the z direction to find out whether the point that we have reached is a minimum or not. To better visualize this we draw what are called contour plots which I show on the right hand side of this picture. So, think about a plane that cuts this objective function plot parallel to the x_1, x_2 surface. So, for example, let us say you think of a plane like this which is parallel to this and it is going to cut the objective function plot.

Now, if you have a plane that is parallel to the x_1, x_2 surface then what we are going to see is we are going to have the objective function

value be a constant across the plane because when you project it here it is going to be at a particular $f(x_1, x_2)$ value or z value. So, what one could say then is that if I cut this surface with the plane parallel to x_1, x_2 surface then I am going to get what are called contours on the x_1, x_2 surface. So, you want to think about it this way. So, here is a plane that is cutting the surface. So, on the plane wherever the surface is cut you are going to have a contour and what we are going to do is we are going to project that contour onto this x_1, x_2 surface. So, that is the plot here. So, for example, we could take $z = 5$ and then have that plane cut this surface then let us see what the projection of that in x_1, x_2 axis will be.

So, we know that we are going to keep or hold the $z = 5$ as a constant. So, you will get this equation root of $x_1^2 + x_2^2 = 5$ which will give you $x_1^2 + x_2^2 = 5^2$. So, this we all realize as equation of a circle centered at the origin with a radius of 5, which is what you see in this plot. Similarly if you say $k = 4$ you will get this contour plot and $k = 3, k = 2$ and so on.

Now, an interesting thing to notice is if I start with some decision variable values here and let us say I want to improve my objective function, I know that if I pick a contour like this and then from here if I keep moving on this contour I am not going to make any improvement to my objective function value. I also know that as I go away from this point in this direction, let us say I go to a new point, then that would be on a contour where the value of k is larger than where I was here. So that means, I have increased my objective function. So, if I move in any of these directions I am going to increase my objective function.

So, the one way in which I should move to decrease my objective function is to move in this direction because however, much I decide to move if I let us say I move here then this is the contour. So, the objective function value on this contour is less than this contour. So, this point is a better point than this one.

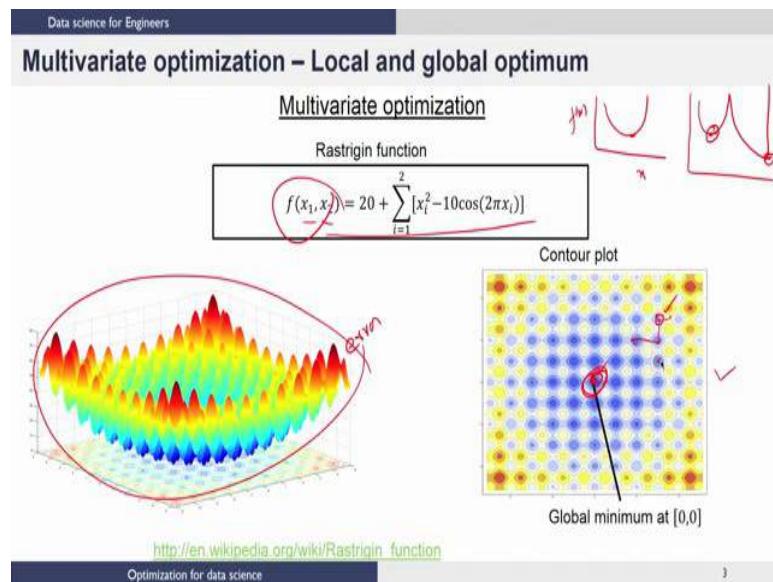
So, this is the basic idea about how we optimize this function.

Now, as I speak you would have noticed that there are two decisions that I need to make. The one decision that I need to make is of all of these directions, what direction should I choose? So, I have to sit here and then make a choice about the direction that I need to figure out. And once I choose a particular direction let us say I choose this direction how far should I go in this direction is another decision I should make.

So, for example, if I go here I have made some improvement to my objective function let us say if I go here I have made much more improvement to my objective function, but let us say if I go here I have

actually made my objective function worse. So, there are two important things that I need to decide, one is the direction in which I should move in the decision variable surface and once I figure out which direction I should move in how much should I move in the direction. So, those are two important questions that we need to answer.

(Refer Slide Time: 09:59)



We will answer these questions when we look at numerical methods of solving these kinds of unconstrained optimization problems. In this lecture what we are going to do is we are going to show you the analytical conditions for minimum in a multivariate problem. Before we do that just like we saw in the univariate case, let us say this is $f(x)$ x and then I have a function like this which has only one minimum which is a global minimum and then I also showed another case where we have a function may be like this where this is one minimum this is one minimum both are minima this is a local minimum and this is a global minimum.

This same thing happens in the multivariate case also. Here is an interesting example of a function where there are two decision variables let us say x_1 and x_2 and the function is of this form. If you plot this in a three d plot you will get this you can see there are many hills and valleys in this and you will notice if we do the projection of this on to the x_1, x_2 surface you will see that there are several minima in this picture and you will see that the global minimum is here.

So, you can see how hard it can become in case of functions like this, where if you let us say, you start from here, then clearly you know one of the good things to do would be to go to this minimum. And you will be here and from here when you look at the conditions

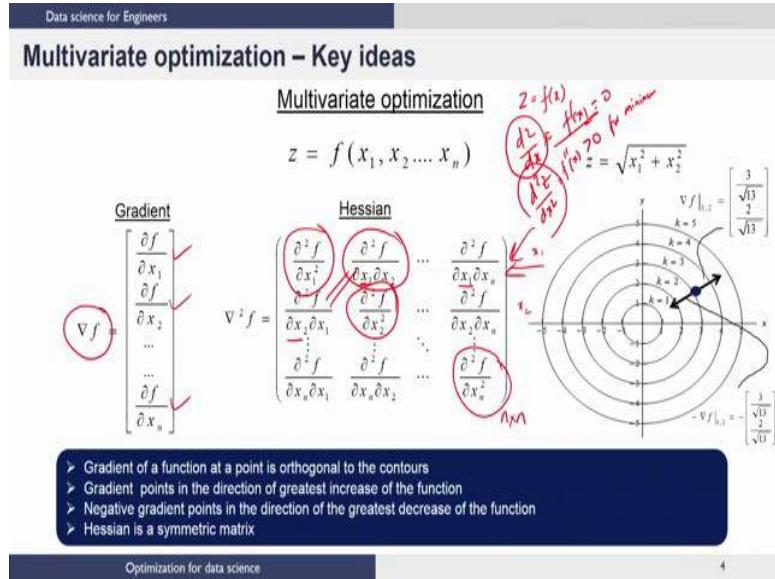
for minimum there will not be any difference between the conditions here and the conditions here in terms of the first order and second order conditions that we talked about in the univariate case we will see what the equivalent conditions are in the multivariate case subsequently.

However, from just those conditions you will not see any difference, nonetheless if you actually compute the objective function value at this point and this point this will be much smaller than this. However, when you are here you have no reason to suspect that a point like this really exists unless you do considerable analysis. So, in cases like this what you will have to do is, you have to see whether you can improve it further, that basically means though you know locally you are very good here you have to do some sacrifice and then try and see whether there are other points which could be better. So, there are algorithms which will let you jump here and then maybe will jump here and so on, but these are all algorithms where it is very difficult in a general case to prove that I will go and hit the global minimum.

Now, remember this is something important to note particularly from a data science viewpoint because let us say this is your error surface. Just as a idea for you to think about how important these concepts are and you are trying to fit a model and the best parameters for the model are here. But a typical optimization algorithm would get stuck anywhere in any of these local optima.

From a data science viewpoint what it means is that the error is not as small as it could be here. However, from the model viewpoint if you were to change the parameters from this value in any direction you change, you will be finding out that the error actually increases in the local region. So, there is very little incentive to improve your objective function value, sorry a very little incentive to move away from this point because locally you are increasing your objective function value. So, ultimately your algorithm might find parameters which while may be acceptable are not the best. So, this is one problem that needs to be solved really to have good efficient data science algorithms.

(Refer Slide Time: 14:54)



So, let us get back to finding out analytically how we solve this problem. So, if you have a multivariate optimization problem where you have z is $f(x_1, x_2, \dots, x_n)$. And in the univariate case let us just contrast this with the univariate case. So, let us say $z = f(x)$ just one variable. Then remember we said the necessary condition for a minimum is that I should have $dz/dx = f'(x) = 0$ and then we said $d^2z/dx^2 = f''(x) > 0$ for minimum. So, these are the conditions that we described in the previous lecture.

So, the derivative in a single dimensional case becomes what we call as a gradient in the multivariate case. So, in this case we have dz/dx or df/dx . However since there are many variables we have many partial derivatives and the gradient of the function f is a vector such that in each component I compute the derivative of the function with respect to the corresponding variable. So, for example, this is $\partial f / \partial x_1$ is the first component $\partial f / \partial x_2$ is the second component and $\partial f / \partial x_n$ is the last component. So, this replaces this in the single variable case.

And this is replaced by what we call as a hessian matrix in the multivariate case. So, this is a matrix of dimension n by n . The first component is $\partial^2 f / \partial x_1^2$, the second component is $\partial^2 f / \partial x_1 \partial x_2$ and so on, and you fill out the row like this. So, the notation that we have used is we have used the x_1 in the front here and x_2 here for second column, x_3 here for the third column and so on, x_n here for the last column.

Now, this is with respect to variable x_1 . Now, you can do the same thing with respect to variable x_2 . Notice here that x_2 has come before x_1 because it is in the second row and this diagonally is always with respect to the same variable differentiated twice. So, this is $\partial^2 f / \partial x_2^2$.

$f/\partial x_n^2$ and so on. Also notice that this hessian will be a symmetric matrix because for most functions this = this and similarly you will have $\partial^2 f / \partial x_1 \partial x_3$ the next term here will be $\partial^2 f / \partial x_3, \partial x_1$ which will be the same. So, the hessian matrix is going to be symmetric. Remember in the linear algebra lecture we said that we will be seeing symmetric matrices quite a bit and here is a symmetric matrix that is of importance from an optimization viewpoint.

Now, what we need to do is we need to see how these conditions in the univariate case translate to the multivariate case.

(Refer Slide Time: 18:36)

So, much like what we did in the univariate case, we are going to do a Taylor series approximation and what I have done here is I have just written it till two terms there are more terms here. But what we are going to do is we are going to make the argument that if you make the distance between the point that you are at and the next point that you are going to choose very small, then whatever is the leading term in the sum is going to decide the sign of the whole sum.

So, in other words if you take this whole thing right here if you keep making this as small as possible or as small as needed then what will happen is, the fact that whether this infinite sum is positive or negative can be identified only by the first term and if that is positive then the whole sum series sum is going to be positive and so on. So, that is the kind of logic that we are going to use again here.

Much like before we said that if I keep making this small I need to only look at this here and much like the univariate case if this does not go to 0, I can make this term either positive or negative. To see this if I take a particular direction and then say $\delta f^T \alpha$. If this turns out to be

negative this number, then if I go in the opposite direction of $-\alpha$ I will get $\nabla(\alpha)$ this will be positive; that means, that I will have a point here which can be either larger than this or smaller than this and if I can find a point such that this is smaller than this then this cannot be a minimum.

So, whatever you do unless this goes to 0, I cannot ensure that this is a minimum point. So, the first condition that we will get is that this is 0. And once that is 0 then I am left with the just this term right here and if you notice this term is of the form $\delta^T H \delta$. We know that this is a symmetric matrix and let us also make sure we understand this clearly the function f is a scalar function and you can see that here also H is an n by n matrix here λ^T will be 1 by n and λ will be n by 1. So, when I do this I will get one by one which is a scalar.

Now, irrespective of this \bar{x} , if this is a positive number then we can say irrespective of whatever direction you take this will always be greater than this in the local region which would qualify this point x^* as a minimum point. So, that is the important idea that that you should remember.

(Refer Slide Time: 22:15)

Data science for Engineers

Multivariate optimization – Summary of conditions

Multivariate optimization

$$(\bar{x} - \bar{x}^*)^T \nabla^2 f(\bar{x}^*) (\bar{x} - \bar{x}^*) > 0$$

$$(\bar{v})^T \nabla^2 f(\bar{x}^*) (\bar{v}) > 0$$

\Downarrow

$\delta^T H \delta > 0$
 $H = \nabla^2 f$
 H is a positive definite matrix
 $\lambda_1, \lambda_2, \dots, \lambda_n > 0$

Condition for Hessian to be positive definite

Hessian matrix is said to be positive definite at a point if all the eigen values of the Hessian matrix are positive

So, we come back to this in the last slide I wrote this as $\delta^T H \delta > 0$, H is symmetric. H is basically this second derivative matrix. Now, we did not see this in the linear algebra lectures, but if I need this condition to be satisfied irrespective of whatever δ is then we call this H as a positive definite matrix. So, if H is positive definite then this will be greater than 0 for all $\delta \neq 0$, clearly when $\delta = 0$ this will be = 0.

So, how do I check if a matrix that I compute is positive definite or not. Remember from the linear algebra lecture we said if I have a symmetric matrix, then I will have the eigenvalues as being real. So, symmetric matrices always have real eigenvalues and the eigenvalues could be positive or negative in this case. Now, the linear algebraic result for positive definite matrix is that if this matrix has let us say n eigenvalues, and if all of these eigenvalues are greater than 0 then this matrix is called positive definite.

In other words if all the eigenvalues of this matrix are greater than 0, it is automatically guaranteed that whenever we compute this for any δ direction we will always get a positive quantity. So, this has already been proved. So, if you want this to be positive for any direction why do we want this we want this because we want $f(x^*)$ to be the lowest value in its neighborhood and that we said will happen if this is positive for any δ or for every δ this should be positive. That condition can be translated to H being positive definite and H being positive definite can be translated to the condition that λ_1 to λ_n the n eigenvalues of H are strictly greater than 0.

Now, what this does is the following. So, in a multivariate case it gives us a way to identify points that could be optimum points and once we identify those points we can compute this hessian matrix at those points and then computation of the eigenvalues of this hessian matrix would allow us to determine whether the point is a maximum point or a minimum point and so on. So, this is the complete equivalent of what we did in the univariate case.

(Refer Slide Time: 25:38)

Data science for Engineers

Overall Summary – Univariate and multivariate local optimum conditions

Multivariate optimization

$\min_x f(x)$ $x \in R^n$	$\min_{\bar{x}} f(\bar{x})$ $\bar{x} \in R^n$
<u>Necessary condition for x^* to be the minimizer</u> $f'(x^*) = 0$ <u>Sufficient condition</u> $f''(x^*) > 0$	<u>Necessary condition for \bar{x}^* to be the minimizer</u> $\nabla f(\bar{x}^*) = 0$ <u>Sufficient condition</u> $\nabla^2 f(\bar{x}^*)$ has to be positive definite

Optimization for data science

So, to summarize in the univariate case the two conditions are f prime has to be zero and f double prime has to be greater than 0. In the multivariate case these translate to $\nabla f = 0$ and the Hessian matrix being positive definite.

(Refer Slide Time: 25:57)

Data science for Engineers

Multivariate optimization – Numerical example

Multivariate optimization

$$\min_{x_1, x_2} x_1 + 2\sqrt{x_2} + 4x_1^2 - x_1x_2 + 2x_2^2$$

First order condition	Second order condition
$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 + 8x_1 - x_2 \\ 2 - x_1 + 4x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ solving $\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.19 \\ -0.54 \end{bmatrix} \checkmark$	$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 8 & -1 \\ -1 & 4 \end{bmatrix}$ $\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 3.76 \\ 8.23 \end{bmatrix} \checkmark$

$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} -0.19 \\ -0.54 \end{bmatrix}$

Optimization for data science

Let us take a very very simple example and identify an optimum solution.

So, consider this multivariate example. So, there are two decision variables and this is a function in terms of these two decision variables.

So, what you can do is you can first construct this ∇f vector which is $\partial f / \partial x_1$. So, that would be $\partial x_1 / \partial x_1$ will be 1, this will be a 0 term this will be 4 time 2 times x_1 , 8 x_1 this would be $-x_2$ and this would be 0. So, the first term is $\partial f / \partial x_1$ here similarly when you do $\partial f / \partial x_2$, I will have a term corresponding to this two. This when differentiated with respect to x_2 will go to 0 corresponding to this I will have a $-x_1$. Now, and corresponding to this I left 4 x_2 which is what we have here.

So, we have these two equations that we need to solve. So, when we solve this and get let us say one of the solutions $x_1^* x_2^*$ is this here. I can check whether this is a maximum point or a minimum point, to do that what I have to do is I have to do this second derivative matrix. So, the way you do the second derivative matrix is the following. So, the first term is $\partial^2 f / \partial x_1^2$. So, we already have $\partial f / \partial x_1$.

So, if you differentiate it with respect to x_1 you will get this term. So, the only term remaining will be 8 which is what we see here and when we look at this we already have $\partial f / \partial x_2$. So, we have to

differentiate this with respect to x_1 . So, the only term remaining will be -1 which will be here and I already told you this is a symmetric matrix. So, you can simply fill in the -1 here and to get this term I already have $\partial f / \partial x_2$ here I differentiate this again with respect to x_2 . So, the only thing that will be remaining would be 4 which is here. So, I have this. Now, what I need to do is I need to compute the eigenvalues for this and when I compute the eigenvalues for this I find the eigenvalues to be both positive, that means, that this is a minimum point.

Now, when we look at this equation here there are two equations in two variables and both are linear equations. So, there is going to be only one solution here and it turns out that that solution is a minimum for this function. So, this finishes our lecture on multivariate optimization in the unconstrained case.

What we will do in the lectures that follow, we will look at some numerical methods for solving these types of problems. We will introduce the notions of how to solve these problems when there are constraints. We look at two types of constraints, one are what we call as equality constraints the other type of constraints are inequality constraints. So, we will pick up from here in the next lecture.

Thank you.

Data Science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 25
Nonlinear Optimization Unconstrained Multivariate Optimization

In this lecture we will continue with Unconstrained Multivariate Optimization. In the last lecture we saw the conditions for a point to be an optimum point a minimum point for multivariate functions.

We described multivariate functions as functions with several decision variables. What we are going to do in this lecture is to look at the same conditions and show how you can numerically solve these optimization problems. And the reason why we are teaching this in a data science course is the following, if you think of any data science algorithm, you can think of it as some form of an optimization algorithm and the techniques that you will see in today's class are also used in solution to those data science problems or data science algorithms. And you will see these numerical methods as what is called as learning rule in machine learning and so on.

(Refer Slide Time: 01:24)

The slide is titled "Unconstrained multivariate optimization - Directional search". It contains a bulleted list of points and a diagram of a mountain peak with labeled points. A photograph of a hiker on a snowy mountain is also included.

Unconstrained multivariate optimization - Directional search

- Aim is to reach the bottom most region
- Directions of descent
- Steepest descent
- Sometimes we might even want to climb the mountain for better prospects to get down further

global maximum
saddle point
local maximum
global minimum
local minimum

global maximum
saddle point
local maximum
global minimum
local minimum

Optimization for data science

So, let us look at an unconstrained multivariate optimization problem. In unconstrained multivariate optimization problems we are

going to solve these using what we call as a directional search. The idea here is the following, if you are on the top of a mountain keying and you are interested in reaching the bottom most point from where you are pictorially shown through this picture here, you will see that there are several different points in this surface. This is a point which is at the bottom of the hill. So, we call this as a minimum point.

However, this is a local minimum because right next to it there is another point which is even lower which we call as the global minimum. We also see that there is a local maximum here a global maximum here and there are also points such as these which are called the saddle points. When you look at optimization algorithms, the aim is for us to reach this point. We want to avoid points like this because as I described before when we reach these kinds of regions while locally we cannot make our algorithm find anything better we know that globally this is not the best. So, in that sense we could do better.

Nonetheless this is an OK point if it is not very far in terms of the performance from a global minimum. However, we want to avoid points like these saddle points and so on, because as you see even in this picture you know saddle points could be very far away from the actual solution.

So, the aim is to reach the bottom most region. Typically what you would do is the following. So, if you are at a particular point here and then you say look let me go to the bottom of the hill as fast as possible, then you would look around and find the direction where you will go down the fastest. So, this is a direction we call as the steepest descent. So, the direction in which I can go down really fast and I will find that direction and then go down the direction.

Now, the way optimization algorithms work is the following, you are at a point you find the steepest descent direction, and then what you do is you keep going in that direction till a sensible amount of time or in this case the length of the step that you take in that direction.

The reason for this is the following, the reason is you could find this as the steepest descent direction and you could keep going in this direction, but let us say beyond this point you really do not know whether this is going to be the steepest descent at that point also. In other words is this going to continue to be the steepest descent till I get to my best solution. Now, that is something that is you cannot guarantee easily.

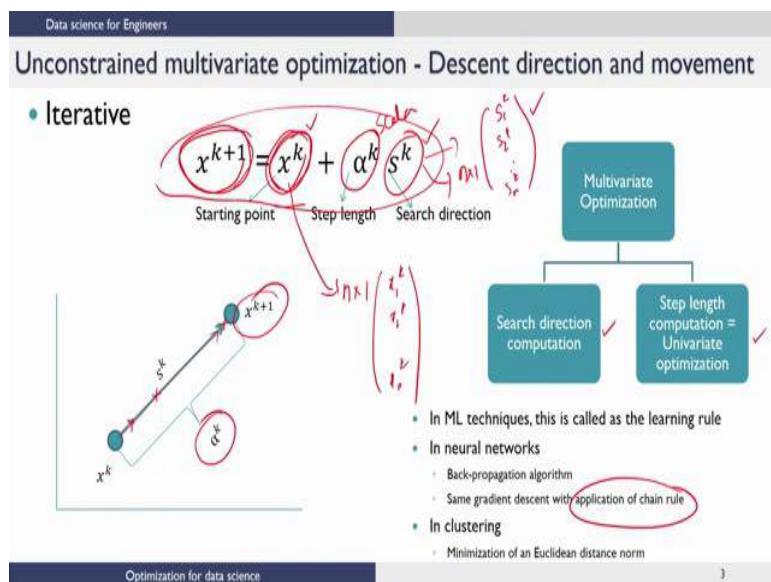
So, smarter strategy would be to find the steepest direction at wherever you are and then find how long you are going to move along this direction, go to the next point in the direction and then at that point reevaluate all the directions, and then find new steepest descent direction. If it turns out that the direction that you are on is continuing to be the steepest descent direction you continue to go on that

direction, if not you find a new direction and then go in the direction. So, that is a basic idea of all steepest descent algorithms.

Now, notice that you know we do the steepest descent and let us say we end up here, then at that point you will find no direction where you can improve your objective function that is you cannot minimize your objective function anymore. In which case you are stuck in the local minimum. So, there are optimization algorithms which, when they try to get out of the local minimum. The only way to do it is let us say if you are here the only way to get to global minimum is to really climb up a little more and then find directions and maybe you will find another direction which takes you here.

In some cases we might construct these optimization algorithms in such a way that you might actually make your objective function worse in the interim, looking for better solutions than your local optimum solution. So, that is what we have written here sometimes we might even climb the mountain to get better perspective. So, for example, if you are here and then say if you look at this you are going to land up here maybe you can go in this direction climb up actually get a better perspective and then come down here. So, some of these are mathematically formulated and solved. So, the basic idea of how these algorithms work is explained in this slide.

(Refer Slide Time: 06:50)



Now, let us see the mathematics behind whatever I described in the previous slide. So, the current point that I am at is what I would call as x_k . So, k stands for the k th iteration. So, at the 0^{th} iteration you will start with x_0 , move to a point x_1 and at the first iteration you will start at

x_1 and move to x_2 and so on, till you find the solution that you are happy with.

The discussion that we had in the previous slide is seen here mathematically. So, what we are interested in is finding a new point which is better than x generally. And the notion of steepest descent is the following or for that matter any other search direction is the following. So, I have this point and I am going to move in a direction that I have chosen. If I choose the steepest descent then I choose that direction, if it is some other direction we choose that direction.

And then what I am going to say is I have a current point I have chosen a direction in which I want to move the only other thing that I need to figure out is how much should I move in this direction, and remember from vectors we talked about in the linear algebra portion of this course. If I start from x_k and then keep moving in the direction of s_k this is what it will be I will be on this line. Now, the question is this x_{k+1} where is it placed. So, if α is very small it will be here slightly bigger x_{k+1} will be here even bigger x_{k+1} will be here and so on. So, how do I find this step length that I need to take in this direction.

So, notice that something interesting has happened. If I am in a particular point. So, remember this is also vector because we are talking about multivariate problems. So, there are several decision variables. So, this will be an n by 1 vector if I have n decision variables. So, this could be something like x_{1k}, x_{2k} all the way up to x_{nk} and these are the current values for variables x_1, x_2 all the way up to x_n which are the decision variables that we are trying to find. Now, for this equation to be correct this is also going to be n by 1 vector the search direction and this is essentially a scalar this is just a number that we need to find out which is a step length.

Now, we will show you what the steepest descent direction is, but you figure out this direction somehow. So, this is going to be a direction vector. So, this direction vector will be something like s_{1k}, s_{2k} all the way up to s_{nk} . So, let us assume that we have somehow figured this out, then the real question is what is the step length. So, one idea is to figure out the step length, so that this when substituted into the objective function is an optimum in some sense.

So, that is what we are going to try and do. So, the key take away from this is that if you are at a current point which you know and if you somehow figure out a search direction, then the only thing that you need to then calculate is the step length. And since step length is a scalar what happens is a multivariate optimization problem has been broken down into a search direction computation and finding the best step length in that direction which is a univariate optimization because we are looking for a scalar α .

In general this kind of equation that we see here you will see in many places as we look at machine learning algorithms in clustering, in neural networks and many other places, in machine learning techniques this is called the learning rule. Why is it called the learning rule? It is called the learning rule because you are at a point here and you are going to a new point you are learning to go to a point which is better than wherever you are and I mentioned to you before that we could think of this machine learning algorithms as being optimization problems solutions to optimization problems.

So, if you talk about neural networks one of the well known algorithms is what is called a back propagation algorithm. It turns out that the back propagation algorithm is nothing but the same gradient descent algorithm. However, because of the network and several layers in the network it is basically gradient descent we are including an application of a chain rule which we all know from our high school. Similarly in clustering algorithms you would see that clustering algorithms would turn out to be minimization of a Euclidean distance now.

(Refer Slide Time: 12:16)

Data science for Engineers

Steepest descent and optimum step size

- Minimize $f(x_1, x_2, \dots, x_n) = f(x)$
- Steepest descent**
 - At iteration k starting point is x^k
 - Search direction $s^k = \text{Negative of gradient of } f(x) = -\nabla f(x^k)$
 - New point is $x^{k+1} = x^k + \alpha^k s^k$ where α^k is the value of α for which $f(x^{k+1}) = f(\alpha) = \text{is a minimum (univariate minimization)}$

So, let us now, focus on the steepest descent and the optimum step size that we need to take. So, the steepest descent algorithm is the following. At iteration k you start at a point x_k . Remember with all of these optimization algorithms you would have to start with something called an initialization which is x_0 and this is true for your machine learning algorithms also. All of them have to start at some point and depending on where you start, when you go through the sequence of steps in the algorithm you will end up at some point let us

call x^* , and in many cases if the problem is non convex that is there are multiple local minima and global minima the point that you will end up is dependent on not only the algorithm, but also the initial point that you start with.

That is the reason why in some cases if you run the same algorithm many times and if the choice of the initialization is randomized, every time you might get slightly different results. So, to interpret the difference in the results you have to really think about how the initialization is done. So, that is an important thing to remember later when we learn machine learning algorithms.

So, as I said before, we start at this point x_k and then we need to find a search direction and without going into too much detail the steepest descent will turn out to be a search direction s_k which is basically the negative of gradient of $f(x)$, where $f(x)$ is your objective function. So, if $f(x)$ is an objective function of the form with this many decision variables then $\text{grad } f$ is basically $\partial f / \partial x_1$ all the way up to $\partial f / \partial x_n$ and negative $\text{grad } f$ would be this, and we keep this as the search direction $s_k = -\text{grad } f$ and this is called the steepest descent search direction.

The key thing that I really want you guys to notice here is the following, x_k is known, the function is known. So, to get to x_{k+1} , x_k is known since the function is known we know also the grad of f and s_k is given as the $-\nabla(f)$ evaluated at x_k . So, basically this is going to be let us say some functional form - g_1x all the way up to g_nx all you are going to do is simply substitute for this x , x_k . So, that basically gives you the search direction. So, this is given this is calculated once this is given.

Then the only thing that I need to find out is this α_k and the way they are value for α_k is found out is by looking at this $f(x_{k+1})$. Now, substitute this x_{k+1} into this. So, you are going to have $f(x_{k+\alpha_k})$ sk α_k , s_k . In this you know this you know this. So, this f is going to simply become a function of α right. So, let me put α_k here. So, this is going to be a function of α .

Now, in the previous slide we talked about this and we said α is a scalar. So, it is just one variable. So, this becomes a univariate optimization or a univariate minimization problem. So, this is a critical idea that I want you to understand.

Now, any univariate minimization algorithm can be deployed to find out what α_k is. So, you deploy a univariate minimization algorithm find a value for α_k and then plug this back in then you have your algorithm for your multivariate optimization which will go something like this. So, you start with x_0 then x_1 is going to be $x_0 + \alpha_0 s_0$. So, x_0 is given based on your initialization, s_0 is calculated, α_0

is optimized for, then you go on to x_2 is $x_1 + \alpha_1 s_1$ and so on. And you keep doing this till you use some rule for convergence you say at some point the algorithm is converged. So, that point is what I am going to call as x^* .

Connection to machine learning algorithms is the following this α is usually called as the learning rate. You could either optimization find out or you could actually pick a value for this and then say let us run the algorithm let us not optimize for this $\alpha_0 \alpha_1$ and so on, I will give you a fixed value of α , α_{fixed} . So, you simply run your algorithm with fixed value of α which is x_0 x_1 will be $x_0 + \alpha_{\text{fixed}} s_0$, x_2 will be $x_1 + \alpha_{\text{fixed}} x_1$ and so on. So, that is also something that you could do. Nonetheless this is the critical equation which is used to optimize an objective function.

In the next lecture we will look at a numerical example that illustrates some of the ideas that we have seen. Now, see you in the next lecture.

Thanks.

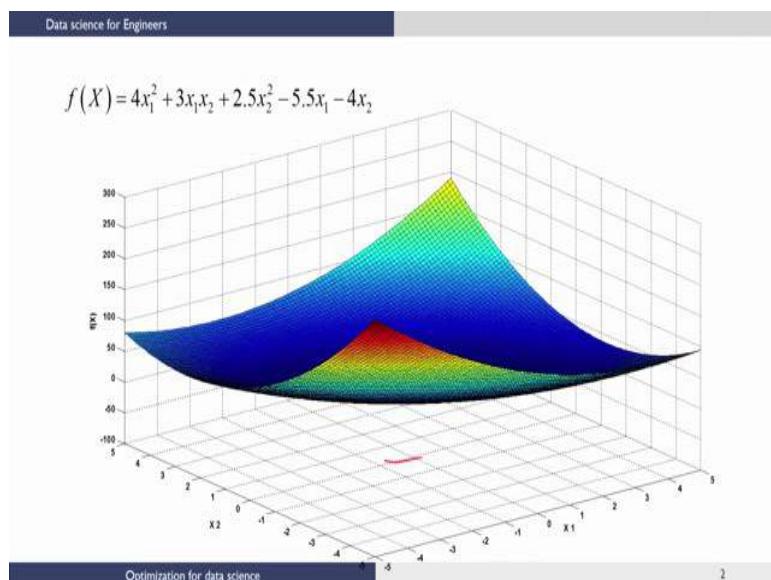
Data Science for Engineers
Prof. Ragunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 26

Numerical Example Gradient (Steepest) Descent (OR) Learning Rule

Let us continue our lectures on optimization for data science. I am going to start out this lecture by showing you a numerical example of how gradient descent works in optimization. In many cases this is also called the learning rule in machine learning algorithms.

(Refer Slide Time: 00:35)

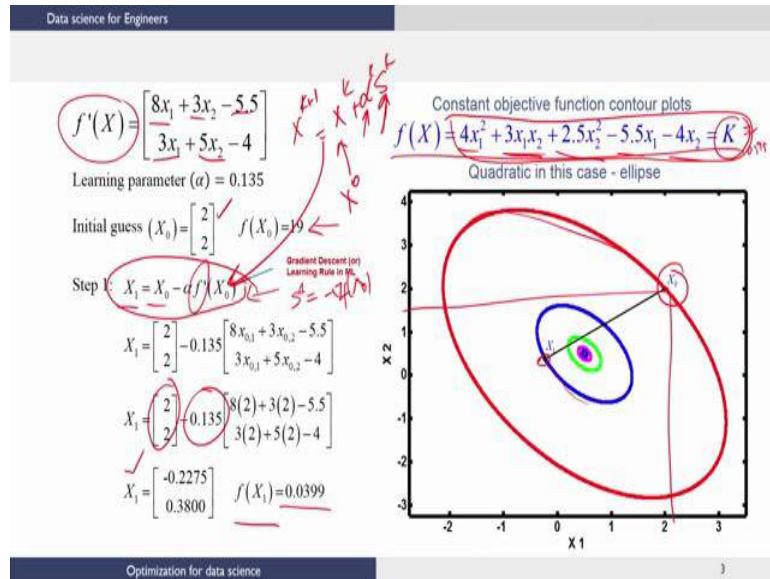


Let us look at a function $f(x)$ which is $4 x_1^2 + 3 x_1 x_2 + 2.4 x_2^2 - 5.5 x_1 - 4 x_2$. We are interested in minimizing this function. As you can notice this is a function of two variables x_1 and x_2 . So, there are two decision variables that we can choose values for, to minimize this function.

As I mentioned before if it was a univariate optimization, then you could visualize the picture in two dimensions with the y axis being the objective function value and the x axis being the decision variable. Now, since we have a function with two variables x_1 and x_2 we visualize this in 3 dimensions, you have the two dimensions x_1 and x_2 on the plane below here and you have the z axis which is $f(x)$ which is

the objective function value. So, we are trying to look at how an algorithm would minimize this function in a numerical fashion.

(Refer Slide Time: 01:44)



Now, notice from the previous lecture we described that while you try to minimize these functions you do what are called contour plots. And if you looked at the previous slide you would notice that x_1 and x_2 are in a plane and the objective function is a value that is projected outside the plane. So, if you think about constant objective function values then what you think about is a constant z value in the previous graph, and a constant z value would be a plane which will be parallel to the plane of the decision variables x_1 and x_2 . So, when that cuts the surface that we saw in the previous graph then you have what are called these contour plots.

So, while we are trying to minimize a function which is represented in the z direction, all the changes to the decision variables are being done in A_2 -dimensional plane which is shown here. For example, each of these curves that we see here are contour plots and as I mentioned before these contour plots are plots where the objective function takes the same value. So, from the previous slide you would notice that if you were to pass a plane through the surface you would see a curve like this would be traced on the surface, and as you move the surface up and down the size of the contour will increase or decrease correspondingly.

Now, the way most optimization problems work is the following. So, let us say we start with some value for x_1 and x_2 . So, we simply guess a value for that and this is what we call as initialization in the optimization. So, let us assume that we initialized this problem at x_0 . So, you would notice that this initialization basically says here is your

value for x_1 and here is a value for x_2 . So, we have picked some x_1, x_2 , and we have initialized this problem.

Now, we know that the constant objective function values are contour plots and if we look at this equation here what we are saying is that the function $f(x)$ which we saw from the previous slide. If we were to find a constant value for this function then I have to set $th = k$. Then I can look at this equation and then say how would this constant contour plot for $f(x)$ look in the x_1, x_2 plane and you would notice that this is quadratic in this case. It is actually going to be an ellipse in for this particular function. So, this ellipse that we trace would be for some particular value of k and our initialization point is here.

Now, the way to interpret this is to say if we were to keep moving on this contour you would make no improvement through your objective function that is your objective function will not decrease because it is a constant contour plot. Now, in gradient descent we wrote this equation where we had $x_{k+1} = x_k + \alpha k s_k$, we said this is the current point, this is the step length and this is the direction in which we should move.

So, let us look at this picture here. When we start we have x_0 which is initialization which is this point. What we need to do is we need to find a direction in which to move and once we find a direction in which we would like to move, then we will find out a learning rule or a learning constant which will take us to the next point. So, initial guess in this case that we have chosen is about 2 2 and $f(x)$ naught value when you substitute this 2 2 is 19.

So, the next step is the following. So, we are going to say x_1 is x_0 . Now, if I take this direction as $-\nabla f$ which is what we discussed last time which I have written here as f' prime then, the equation will be the new point x_1 is $x_0 - \alpha$ times $f'(x_0)$. So, this grad is evaluated at the point that you are currently at. So, the same equation becomes this here. So, just to illustrate the idea of how an optimization approach works we are going to pick some α here, which we have picked as 0.135 here, there is a way in which you can automate this and here we are just using this number.

Now, if I take ∇f , so when I differentiate this function with respect to x_1 , I will get from this term $8x_1$, from this term I will get $3x_2$, and from this term I will get -5.5 . So, $\partial f / \partial x_1$ is going to be this term. And $\partial f / \partial x_2$ I am going to get $3x_1$ from this term, $5x_2$ through this and this -4 I am going to get from here. So, this is $\partial f / \partial x_2$. So, that is your f' prime or ∇f .

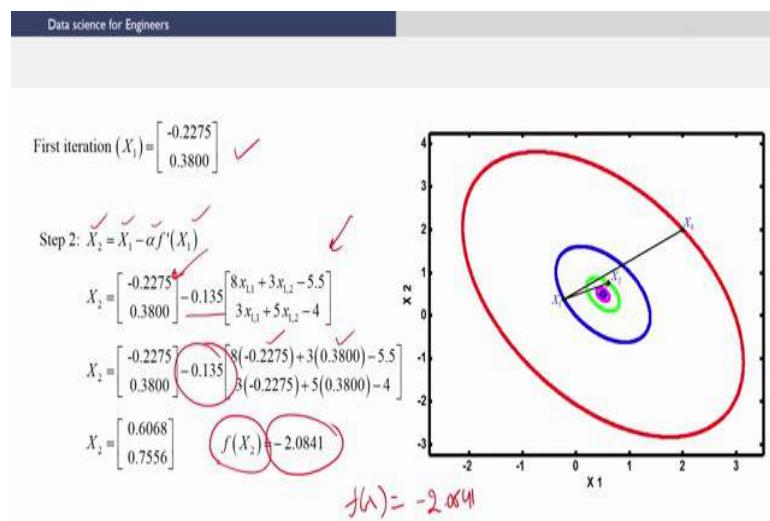
Now, what you need to do is, once you identify this if you look at this equation this direction which is a gradient direction has to be evaluated at x_0 and since our original x_0 is 2 2, I am going to substitute

the value of 2 2 into these equations. So, I have 8 times 2 + 3 times 2 - 5.5, and 3 times 2 + 5 times 2 - 4 and this is the learning parameter and this is our original point which gives me a v point x_1 after simple computation. And when you compute the function value at x_1 you notice that the original function value was 19. Now, it is come down to this number right here.

So, the point that we are trying to make is this direction is actually a good direction and then you move in the direction and you find that your objective function value decreases which is what which was our original intent because we are trying to minimize this function. And as I mentioned before this, gradient descent is usually called the learning rule. So, when you have parameters let us say that you are trying to learn for a particular problem this keeps adjusting this equation keeps adjusting the parameters till it serves some purpose and this adjustment is usually called the learning rule in machine learning.

So, this is the new point that we are right and if you find out this value which is 0.0399 and then set this k to be this number whatever was $f(x)$ 1 which is 0.0399, then you would notice that the equation form remains the same except this constant has changed. So, this is continuing to be an ellipse, but it is an ellipse that are shrunk from your original ellipse. So, the constant contour plot, this blue plot, is the plot at which $f(x)$ will take a value 0.0399. So, wherever you are on this blue curve or the blue contour the objective function value is the same. So, this is a first step of the learning rule that we see.

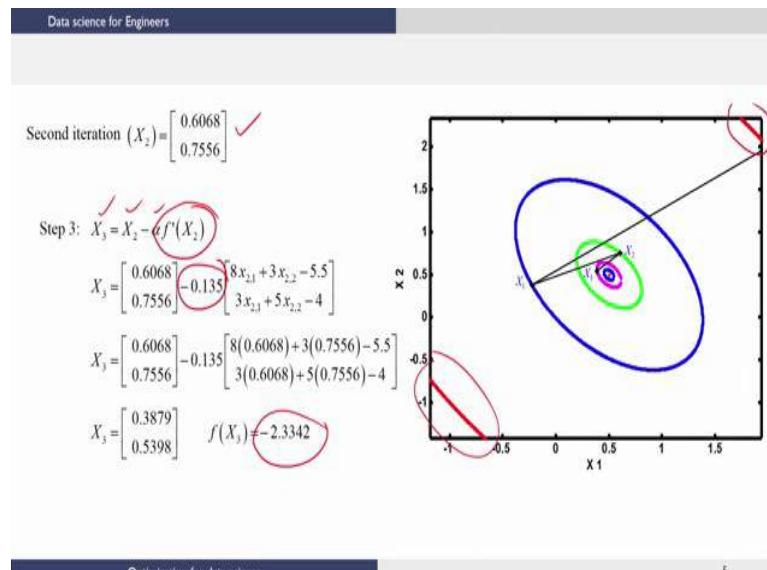
(Refer Slide Time: 09:50)



Now, let us proceed to the next iteration. The next iteration is pretty much exactly the same. What you can see here is now, we start with x_1 which was what we identified from the previous iteration and x_2 is $x_1 - \alpha f'$. So, pretty much we are doing the same thing I substitute the value of x_1 here α remains the same and I do the same $\partial f / \partial x$, but now, I evaluate the gradient at the new point x_1 . So, if you notice here in the last slide we had put 2 and 2 for these values, but in this you would see I am using the x_1 value - 0.2275, 0.3800 and so on. And in this case the learning rate remains constant, but in more sophisticated algorithms or algorithms where you could actually optimize the size of this learning parameter as we go along in the algorithm.

Nonetheless, the ideas are pretty much the same only that this number will keep changing iteration to iteration. Now, we get a new value x_2 and notice that this new value of $f(x_2)$ when substituted into this function $f(x_2)$ give you even smaller objective function value. In fact, the objective function has become negative. So, this new point is shown here as x_2 . Now, again much like how we discussed the previous iteration in the last slide, in this iteration if you were to take the function $f(x)$ and then set it = - 2.0841 then that would be again an elliptical contour and that contour is actually described by this green contour.

(Refer Slide Time: 11:53)



So, one more iteration you can simply follow through the steps same thing

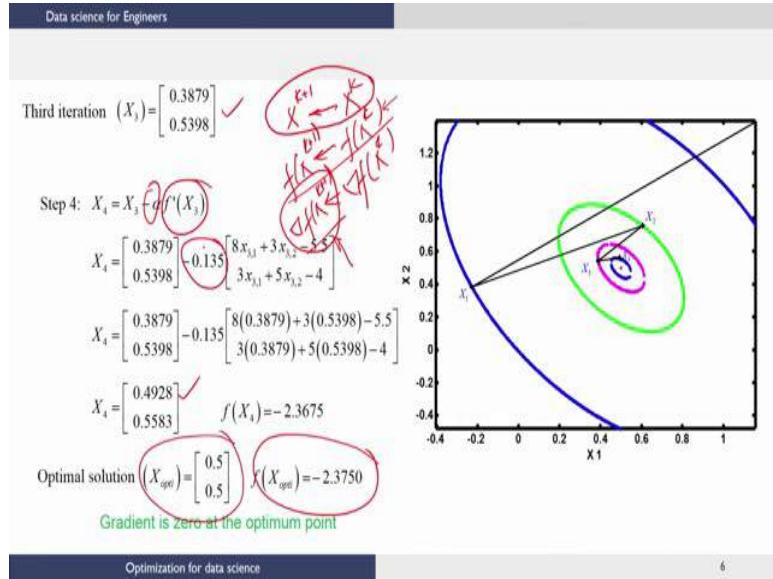
here x_2 from the previous slide the new x_3 value which is $x_2 - \alpha$. Now, again the gradient is evaluated at the new point and the α remains the same and now you notice from the previous slides. Let us go back quickly to the previous slide and see what the value was. The value of the objective function was - 2.0841.

Now, when you look at the new point x_3 the objective function value has decreased even more it has become - 2.3341. So, we notice that at every step of the algorithm the objective function keeps improving for us here in this problem improvement means the objective function value keeps coming down and since at every point and the objective function value keeps coming down our hope is at some point it will hit the minimum value. How do you understand if it is a minimum value or not? It is something that that we will discuss in the next slide.

Nonetheless all I want you to notice is that at every iteration the value of the objective function keeps coming down because we are trying to minimize the objective function, and you can see the kind of improvement you get as we keep zooming down this picture, this was our original red elliptical contour which is kind of going outside the frame. So, this is a big elliptical contour from where we started, but we have improved the objective function to come here.

And just to make the connection between data science and optimization if this objective function were an error in some function that you are approximating your initial error is very large and as we learn or as the machine learns to approximate the function, what it does it keeps improving and it keeps finding out new points which could be the parameter values that that you are trying to find out such that the error keeps decreasing. So, that is what is happening here in this example.

(Refer Slide Time: 14:16)



So, you could go through this process. The third iteration maybe gives us this x_3 and then the next iteration gives you an $f(x_4)$ value which is this and you can notice that the objective function value has come down.

Also notice a couple of other interesting things that we see as we go along through this optimization procedure. If you noticed, I think the first value was very high for $f(x)$ and after the first iteration the objective function value came down quite a bit, and then we have gradually keep improving the objective function value. And as you get closer and closer to the optimum the gradual improvement in your objective function value in the iterations that come later in the algorithm are going to be less. What I mean is I the very first improvement when we went from x_0 to x_1 was a large improvement. But when we go from x_2 to x_3 and x_3 to x_4 , the improvement in the objective function, while the objective function is improving, the improvement amount of improvement keeps decreasing.

And you would expect that because as you get closer and closer to the optimum you know that the optimum point is where the gradient goes to 0. So, as close to the optimum if the function is reasonably continuous then you are going to have derivatives which are very small and if you notice each of these steps, this is one thing that dictates the size of the step you take. And if this is a constant value the size of the step you are going to take is going to come down and also the improvement in your objective function is going to come down. But keep in mind the objective function keeps improving all I am saying is that the amount by which it improves will keep coming down.

So, if you do this for a few more iterations you will get to the optimum point which is this solution 0.5, 0.5 and the function value at this optimum point turns out to be this. Now, the couple of things that I would address here. So, when you write an algorithm like this or when an algorithm like this works you have to tell the machine to stop doing this algorithm at some point. Which is what in optimization terminology called as convergence criteria. And the way the convergence criteria works is the following there are many ways in which you could post a convergence criteria and then say this the algorithm has converged.

So, we talked about the decision various values themselves decision variable values themselves. So, there is let us say we are at x_k and we are finding a new variable x_{k+1} we have this we have also the value of the function at x_k and the value of the function at x_{k+1} , and we have also got the gradient of the function at x_k , and the gradient of the function at x_{k+1} . So, what I am trying to show here is that when I am moving from x_k to x_{k+1} , I could compute all of these quantities. So, you could post a convergence criteria on any of these. So, for example, you could say the difference between this and this, in a vector of different sense, if that is becoming smaller and smaller then you might stop your algorithm.

So, the logic behind this is that if you are making minor modifications to your parameters you can keep doing it to try to get to perfect value, but at some point it starts making not much of a difference. So, you could use this norm as we call it which is the difference between these two values at two different iterations as a condition for saying the algorithm converges. That is when this becomes small enough you say the algorithm has converged.

You could also simply take the difference between the objective function values in two iterations for example. When that becomes very very small you could think about saying that the algorithm has converged. Or you could take the derivative at every point and then when the derivative norm becomes very small you could say the algorithm converges. The logic between these two are that in this case we are saying well we are doing this, but we are not really improving our objective function. So, I am going to be happy with whatever I get at some point and then say if you do not improve significantly and what is significant is something that you define I am going to stop the algorithm.

So, you could do that. Or when you do the norm of this you know ultimately at the optimum value you know the gradient has to be 0, that means, the norm of this vector has to be 0. So, when $\text{grad } f$ becomes very close to 0 then you could say I have converged my algorithm and I am going to stop the algorithm at that point. So, in typical optimization packages or software there are these various options that

you can use to ask for convergence to be detected and the algorithm to stop at that point.

So, this gives you an idea of how the analytical expression that we started with for maximum or minimum is converted into a gradient rule and these are all called as gradient based optimization algorithms. And then we showed you a numerical example of how actually this gradient based optimization algorithm works in practice. We also made the connection between these algorithms and machine learning and as I mentioned before most of the machine learning techniques you can think of them as some form of an optimization algorithm and the gradient descent is one algorithm which is used quite a bit in solving data science problems.

Couple of other things to notice are that the direction for changing your values iteration by iteration, in this case we have taken it as a steepest descent. There are many other ways of doing this you can choose directions in using other ideas that many other algorithms use. So, we here in this introductory course on data science we focused on the most common and the simplest of the search directions which is the negative of the gradient at that point.

And again these algorithms also keep changing the value of the learning parameter or the step length as they would call it in optimization algorithms, iteration to iteration. In this case we have kept that to be a constant just to make sure that we explain the fundamental ideas first before moving on to more complicated concepts. Nonetheless, I just want you to remember that this learning parameter is something that could be changed optimally from iteration to iteration in a given optimization algorithm.

So, with this I hope you have got a reasonable idea of univariate and multi-variate optimization, unconstrained non-linear optimization. What we are going to do in the next lecture is to look at how we can introduce constraints into this formulation, and what effect does a constraint have on the formulation and how do we solve constrained optimization problems. And as I mentioned before these constraints could be of two types, equality constraints and inequality constraints. We will see how we can solve optimization problems with equality constraints and inequality constraints. So, I will see you in the next lecture.

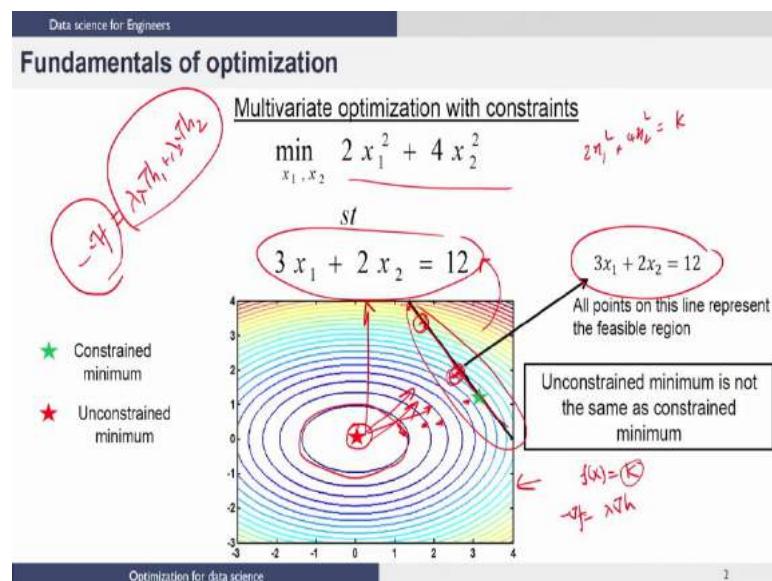
Thank you.

Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 27
Multivariate Optimization with Equality Constraints

Let us continue our lectures on optimization for data science. In this lecture we will look at multivariate optimization non-linear optimization. However, in the previous lectures we saw the unconstrained version of this problem. In this lecture we will look at problems of this type where we have what are called equality constraints. I will explain that is presently.

(Refer Slide Time: 00:46)



And I am going to explain this using a very simple example, but before we dive into this example let us set some context and then ask the question as to why we should be interested in this in a course on data science. The reason why we should be interested in this in a course on data science; the reason why we are interested in constraints in optimization is as I mentioned before, we look at optimization from a data science viewpoint because we are trying to minimize error in many cases, when we try to solve data science problems. And when we minimize error, we said we could use some kind of gradient based algorithm what we called as the learning algorithm to solve the problem.

In some cases while we are trying to optimize or minimize our error or the objective function, we might know some information about the problem that we want to incorporate in the solution. So, if for example, you are trying to uncover relationships between several variables and you do not know how many relationships are there, but you know for sure that certain relationships exist and you know what those relationships are, then when you try to solve the data science problem you would try to constrain your problem to be such that the known relationships are satisfied.

So, that could pose an optimization problem where you have constraints in particular equality constraints and there are several other cases where you might have to look at the constrained version of the problem while one solves data science problems. So, it is important to understand how these problems are solved. Once we look at equality constraint problems we will look at in-equality constraints which is even more relevant. For example, the algorithms for inequality can with inequality constraints are very useful in data science algorithm that is called support vector machines and so on. So we going to look at both equality and inequality constraints.

Now, let us look at this problem right here. So we are interested in mini-mizing the function here which is $2 x_1^2 + 4 x_2^2$ and like we discussed before this is an objective function in 2 variables x_1 and x_2 . So, real visualization of this would be in 3 dimensions where the objective function value is plotted in the z direction. However, we know that we can work with contour plots and if you look at this picture here, you see these contour plots. These are plots where each of this contour is a constant objective function contour and again you would see that these are ellipses because if you look at constant contour plots then you have $2 x_1^2 + 4 x_2^2$ equal k this you will see is an ellipse that is what is plotted here.

Now, if you look at the optimum value for this function and just by inspection you can see that the optimum value is 0 because this are functions where I have terms which are squares of the decision variables. So, there are 2 terms here $2 x_1^2 + 4 x_2^2$ and the lowest value that each one of these could take has to be 0 and that basically means the unconstrained minimum is at $x_1 = 0$ $x_2 = 0$ the objective value at that point is also 0.

So, which is what is shown here in this point right here. So, if I had no constraints I would say the optimum value is 0 and it is at 0 0 the star point here. Now let us see what happens if I introduce a constraint in this case I am going to introduce a very simple linear constraint. So, let us assume that we have a constraint of this form which is $3 x_1 + 2 x_2 = 12$.

Now what this basically means is the following. You are looking for a solution to x_1 and x_2 which also satisfies this constraint that is what it means. So, though I know the very best value for this function is 0 from a minimization viewpoint, I cannot use that value because the 0 point might not satisfy this equation. So, you will notice that if I put $x_1 = 0$, $x_2 = 0$ it does not satisfy this equation. So, the unconstrained solution is not the same as the constrained solution.

Now, let us see how we solve this problem. To understand this we first start by representing this constraint in this 2 dimensional space. So, since this is a linear constraint you will see that it is a line which is here which is what is given by this constraint. Now since we said that whatever solution I get it should satisfy this constraint would translate to saying, I can only pick solutions from this line because only points on this line will satisfy this constraint. Now, you notice that you could pick many many points on this line. So, for example, you could pick this point and the objective function value at that point would be corresponding to a contour which intersects this point. So, if you pick a point here then you would have a corresponding objective function value. Now you could pick a point here then you would see that it would have another objective function value which would be corresponding to the constant function contour that intersects the line at that point.

So, you notice that as you pick different points on this line the objective function takes different values and what we are interested in is the following. Of all the points on this line, I want to pick that one point where the objective function value is the minimum. To understand this let us pick two points and see what happens. So, if I pick a point here and let us say I pick a point here and ask the question which one of these points is better from a minimization of the original function view point. The way to think about this is the following.

So, when I pick a point here I know that the objective function value would be based on the contour which intersects at that point and if you compare these 2 points you will see that this point is worse than this point, from a minimization viewpoint. This is because if you look at these contours these are contours of increasing objective function values and the contour that intersects this point is within the contour that intersects the line at this point. So, basically what that means, is because this is the direction of increasing objective function value the contour intersecting this point is inside the contour intersecting at this point. So, that basically means this function takes a lower value at this point on the line.

So, as you go along the line you see that the value keeps changing and my job is to find that particular point where the objective function value is the minimum. So, this is a basic idea of constrained

optimization solution that we are looking for. The key point to notice here is that the unconstrained minimum is not the same as the constraint minimum. If it turns out that the unconstrained minimum itself is on the constraint line then both would be the same, but in this case we clearly see that the unconstrained minimum is different from the constrained minimum.

(Refer Slide Time: 10:20)

Data science for Engineers

Fundamentals of optimization

Multivariate optimization with equality constraints

At optimum (one equality constraint case)

$$-\nabla f(\bar{x}) = \lambda' \nabla h(\bar{x})$$

$$\begin{pmatrix} -\nabla f \\ \nabla h \end{pmatrix} = \begin{pmatrix} \lambda' \\ 1 \end{pmatrix} = 0$$

In higher dimensions and when there are more than one equality constraint

$$-\nabla f(\bar{x}) = \sum_{i=1}^l [\nabla h_i(\bar{x})] \lambda'_i$$

Gradient lies in the space spanned by the normal of the gradients

So, when I have just only one constraint, how do I solve this problem? I will first give you the result and then explain the result again by going back to the previous slide and then showing you another viewpoint and then how that leads to this solution.

So, if you typically take a multivariate function $f(x)$ in the unconstrained version, let us assume that x is x_1, x_2 and x_n . We know that the optimum solution is $\text{grad } f = 0$ which basically means that $\partial f / \partial x_1, \partial f / \partial x_2$ all the way up to $\partial f / \partial x_n$ equal 0. So, this when I expand this further I will get equations $\partial f / \partial x_1 = 0, \partial f / \partial x_2 = 0, \dots, \partial f / \partial x_n = 0$ and so on.

Now, notice that if there are n variables there will be n equations of this form. So, I have n equations in n variables. So, I can solve for it and find a solution and once I find a solution to find out whether it is a maximum or minimum or a saddle point I calculate the second derivative and then construct the hessian matrix and then depending on whether the hessian matrix is positive definite, negative definite or semi definite and so on we can make judgments about whether the point is a minimum point, maximum point and so on.

Now, let us see what happens if I am trying to solve a problem where I have to minimize $f(x)$ which is $x_1 x_2$ all the way up to x_n . But like I said before let us assume that I also introduced one constraint and I am going to introduce let us say a constraint which is of the form h of $x_1 x_2$ all the way up to $x_n = 0$. So, this is an equality constraint. So, we can always write a constraint in this form, even if you have some number on the right hand side you can always move to the left hand side and write this constraint as something equal to 0. So, basically my job is to find the minimum point for f subject to this constraint. I will just give you the result and then we will see how we get this result.

So, when I want to solve for this problem the result is in this case $\text{grad } f = 0$ itself gave us the result. In this case it will turn out that the result is the following. So, we can write the negative of ∇ has to be equal to some $\lambda \nabla$ of h . So, you can write this as negative or drop the negative and the sign of the value λ will take care of that, but I am writing in this particular form.

So, just to expand this basically says I have something like this I have domain is $\partial f / \partial x_1$ all the way up to $\partial f / \partial x_n = \lambda \partial h / \partial x_1$ all the way up to $\partial h / \partial x_n$. So, if we expand this further I am going to get n equations. In this case I got these equations to be 0 in this case I am going to get equations of the form $-\partial f / \partial x_1 = \lambda \partial h / \partial x_1$ will be one equation, the second equation will be $-\partial f / \partial x_2 = \lambda \partial h / \partial x_2$ and so on, the last equation will be $-\partial f / \partial x_n = \lambda \partial h / \partial x_n$.

So, now notice much like before I have n equations the difference being in the unconstrained case I had zeros on the right hand side in the constrained case I have these terms on the right hand side. Nonetheless these equations are in now $n + 1$ variable because I have my $x_1 x_2$ all the way up to x_n and I have also introduced a new variable λ right here. So, my equations are in $n + 1$ variables. I have only n equations at this point, but notice that I have one more equation that I need to use and that equation is the following if I find some solution x_1 to x_n which satisfies all of these equations.

Then that also has to satisfy the constraint. So, here we are only talking about the gradient form of the various functions, but the equation which represents the constraints also needs to be satisfied by any solution that we get for the constrained optimization problem. So, with these n equations I will also get another equation which is that h of $x_1 x_2 x_n$ has to be $= 0$.

Now, you notice that in this case with one linear constraint I have $n + 1$ equation in $n + 1$ variables. So, I can solve this, so to reiterate the difference between the constrained and unconstrained case was, in the unconstrained case we had n equations in n variables, in the constraint case with just one constraint we have $n + 1$ equations in $n + 1$ variables.

Now, you might ask what happens, if there are more than one constraint there are let us say 2 constraints.

(Refer Slide Time: 16:17)

Fundamentals of optimization

Multivariate optimization with equality constraints

At optimum (one equality constraint case)

$$-\nabla f(\bar{x}) = \lambda^* \nabla h(\bar{x})$$

In higher dimensions and when there are more than one equality constraint

$$-\nabla f(\bar{x}) = \sum_{i=1}^2 [\nabla h_i(\bar{x})] \lambda_i^*$$

Gradient lies in the space spanned by the normal of the gradients

$$\nabla f(\bar{x}) = \lambda^* \nabla h$$

$$f(x) \quad h_1(x) = 0 \quad h_2(x) = 0$$

$$-\nabla f = \lambda_1 \nabla h_1 + \lambda_2 \nabla h_2$$

$$\lambda_1(x) = 0; \lambda_2(x) = 0$$

So, we will see what happens if there are 2 constraints, so in this case we are going to minimize $f(x)$ and now let us say we have 2 constraints we are going to say $h_1(x) = 0$, $h_2(x) = 0$. In this case what is going to happen is the following the solution to the constrained optimization problem is going to be $-\nabla f = \lambda_1 \nabla h_1 + \lambda_2 \nabla h_2$.

So, if you look at this and the one constraint case you will see the similarities. In the one constraint case we introduced one extra parameter, in the 2 constraints case the x introduced 2 extra parameters. In the one constraint case, we simply said $\text{grad } f - \nabla f$ equal $\lambda \nabla h$, in this case this $\lambda \nabla h$ becomes a sum of $\lambda_1 \nabla h_1 + \lambda_2 \nabla h_2$ and so on.

Now, if there are 3 equality constraints, then you would have 3 terms here and if there are 1 equality constraints you will have something like this here where this is sum of 1 terms. So, that part is clear the other part that we need to worry about is if I have enough equations to solve for all the variables when I have 2 constraints, that basically means I have introduced 2 new variables λ_1 and λ_2 . However, this equation irrespective of the number of equality constraints you have will always be n equations.

Because $\text{grad } f$ would be $\partial f / \partial x_1 \partial f / \partial x_2 \dots \partial f / \partial x_n$ so these are all n by 1 vectors. So, this will just give me n equations. Nonetheless if I had 2 equality constraints I need to find the 2 extra equations which are

directly given by the constraints. So, since the optimum point has to satisfy both the constraints, I get one extra equation $x_1 x = 0$ and the other extra equation is $2 x = 0$. So, if you have 2 equality constraints in n variables I will have $n + 2$ variables and $n + 2$ equations and we will always find this to be true because if I had 3 equality constraints, I will have $n + 3$ variables which would be x_1 to x_n , $\lambda_1, \lambda_2, \lambda_3$ and this gradient equation will always give me n equations and the 3 extra constraint would have given me the 3 extra equations.

So, I will have $n + 3$ equations and $n + 3$ variables which can be directly generalized to this form where I have 1 equality constraints. So, let us see in the single constraint case how we get an expression like this that that would be easy to see in the single constraint case. This expression where we have the sum of these terms is slightly more complicated to understand. I am not going to do that. I am going to give you an intuitive feel for why this is true in the single equation case and once you understand that with a little bit more effort you should be able to think about why it should be true for more equality constraints and so on.

So, we go back to the previous slide and when I was discussing this slide and explaining how equality constraints affect the optimum solution. I said there are many points on this line which could all be feasible solutions. Feasible solutions meaning those are all solutions which can satisfy this equation. Nonetheless, there are some points out of those or one point which would give me the lowest objective function value. So, when we looked at candidate solutions for this optimization problem we were looking at the points on this line that that we are interested in because that is a constraint.

Now, let us take a slightly different viewpoints and then look at the same problem from an objective function viewpoint. Now from an objective function viewpoint if you did not constrain me at all and you said you could do anything you want, then I would pick this point as a solution. Now when you look at this point and then say well this is a best point I have let me find out whether it satisfies my constraint and then you substitute this point into this and then you figure out it does not satisfy the constraint.

So, you say look I have to do something because I am forced to satisfy the constraint. So, you will say let me lose a little bit in terms of an objective function perspective and then see whether I can meet the constraint.

So, when we say I want to lose a little bit basically you know as we mentioned before these are contours where the objective function value increases and those are actually not good from a minimization viewpoint. So, while I am here since the constraint is not satisfied, I am willing to lose a little to see whether I can satisfy the constraint and maybe I go to a point here and then this is a constant objective function

contour point. So, if I am willing to give up something that basically means I am going and sitting on different points on this contours and as I am pushed away and away from this minimum point I am losing more and more in terms of the objective function value. That is I am increasing the objective function value. Now logically if you keep extending this argument you will see that, let us say this is the first point I moved here which is basically worse than this because you see this is a contour which is going to be outside of this contour, but I moved here I made my objective function worse, but I still am not satisfying the constraint. So, I give some more I come to this point and I see a contour and this point is worse than this because this contour is outside this contour and if I extract this argument let us say I keep making things worse and the only reason I am making these worse is because I am forced to satisfy this equality constraint.

So, I come up to let us say here and this is still a contour which is much worse than my original solution, but it is still not enough for me to satisfy my constraint. So, if you keep repeating this process, you are going to find a contour here where I touch this line for the first time. So, when I touch this line for the first time is the point at which for the first time, when I give up my objective functions value, I am also able to satisfy the constraint. Now once I find a contour like that which touches this line then there is no incentive for me to go further beyond because going further beyond would mean I would be making my objective function worser.

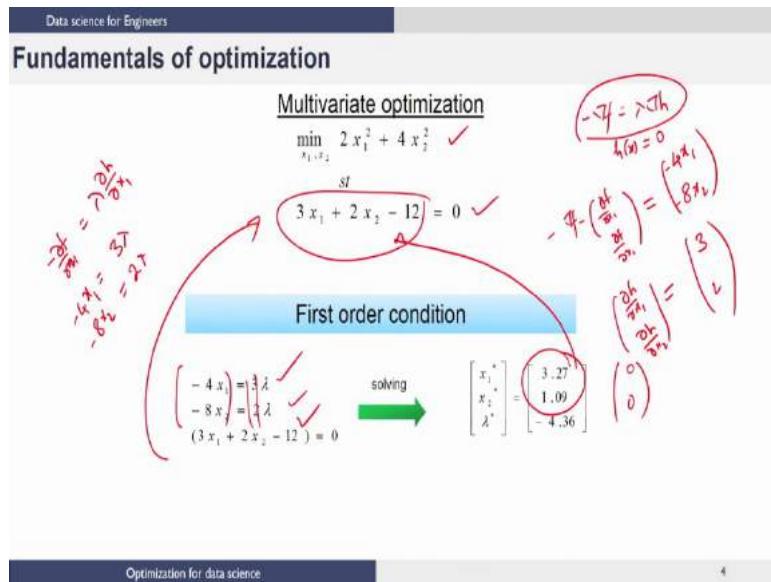
So, when I just touched that line that is the best compromise because that is where I become feasible for the first time and going any further would only make my objective function worse. So, geometrically what this would mean is I keep making my objective function worse till a contour just touches this line. So, at that point this line will become a tangent to that contour and remember what that contour is that contour is $f(x) = k$ for some k

So, we have to choose a k in such a way that this contour, when I have the constraint, the constraint becomes a tangent to this contour and optimum point. This geometric fact when you represent it as equations, you would get the form that I showed which is - grad of f is λ times grad of h . So, this is for the one constraint case, now when you have many equality constraints while it is not as easy to see that as it is in the single equality constraint case the statement that - grad f let us say if I have 2 is λ_1 grad $h_1 + \lambda_2$ grad h_2 this statement basically says that this gradient negative of the gradient has to be written as a linear combination of grad of h_1 and grad of h_2 and that has a geometric interpretation.

So, that is the reason why we get these conditions for optimization with equality constraints the key point that I want you to remember is that as you have more and more equality constraints you will have to

introduce more and more parameters $\lambda_1 \lambda_2$ and so on. However, there will always be enough equations and variables.

(Refer Slide Time: 26:47)



So, let us look at a numerical example to bring all the ideas together. We are going to use the same example that we had looked at in the previous slides. So, we are trying to minimize $2 x_1^2 + 4 x_2^2$ as objective function subject to this constraint. So, we wrote the following equations for the optimum point for identifying the optimum point we said - ∇f has to be $\lambda \nabla h$ and then we have $h(x) = 0$. So, this is what we wrote let us do the computation so that we understand this.

So to calculate ∇f which is basically $\partial f / \partial x_1 \partial f / \partial x_2$. So, if you look at this objective function. So, $f \partial x_1$ will be simply $4 x_1$ and $\partial f / \partial x_2$ will be $8 x_2$. So, we have ∇f so negative grad of f would be negative. Let us look at grad of h . So, h is this equation so if I do $\partial h / \partial x_1 \partial h / \partial x_2$. So, this is going to be $\partial h / \partial x_1$ will be simply 3 and $\partial h / \partial x_2$ will be simply 2 so we have this.

So, now, let us see what this equation becomes. So, this equation you would see is if you put this in a bracket and you will see this easily. So, you have the first equation is $-\partial f / \partial x_1 = \lambda \partial h / \partial x_1$ and we have $-4 x_1 = 3 \lambda$, similarly the second equation would turn out to be $-8 x_2 = 2 \lambda$ and this equation is basically the same equation as the constraint equation that we have here. And as we mentioned before we thought the constraint that would have been 2 variables and you would have got 2 equations which would have been $\nabla f = 0$.

But with an equality constraint we have added a new parameter λ . So, we need 3 equations in x_1 , x_2 and λ we do have these 3 equations here and when you solve these 3 equations you will get this solution and this is your optimum solution in the constrained case which is different from the optimum solution in the unconstrained case which would have been 00. So, in other words we have given up on the value of the objective function.

However, this is a point which would satisfy this equation of the line. So, that is how we deal with equality constraints in an optimization problems. I already described how these are useful or why we should study them in the first place from a data science perspective. With this I will conclude this lecture and in the next lecture we will look at how we handle inequality constraints and I will also explain why we are interested in understanding how optimization problems are solved with inequality constraints from a data science perspective.

Thank you and I look forward to seeing you again in the next lecture.

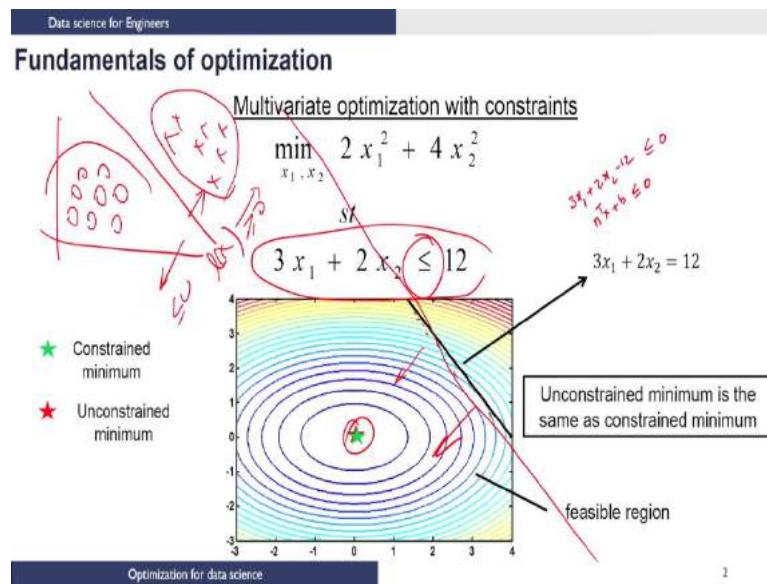
Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer and Science Engineering
Indian Institute of Technology, Madras

Lecture – 28
Multivariate Optimization with Inequality Constraints

We come to the last topic in this series of lectures on optimization for data science. Till now we saw how to solve unconstrained multivariate optimization problems. Then we saw how to solve multivariate optimization problems with equality constraints. Now we move on to multivariate optimization problems with inequality constraints. Before I start explaining some of the key ideas in these types of problems, let us look at why we might need this optimization technique in data science.

Remember in one of the earlier lectures I talked about data for people who like south Indian restaurants and so on.

(Refer Slide Time: 01:07)



So, if you remember that example I said there are a group of people who might like certain type of restaurant there might be a group of people who do not like that kind of restaurant and so on. Then if we were to do a classifier which probably is a line like this then, as we are

trying to solve an optimization problem to identify a classifier like this, we have to impose the constraint that all these data points will have to be on one side of the line and all of these data points will have to be on the other side of line and from our lecture on half spaces and hyper planes in linear algebra we know that if this equation is something like this which is linear equation, then it might be that if the normal is in this direction then this direction is said that if I substitute a value of this point into this it is greater than equal to 0 and on this side it is less than equal to 0 with 0 being the line.

So, now, notice that for each point if we were to write the condition in terms of the equation of the line and then you would see that these become inequality constraints. So, there may be as many inequality constraints there are points and so on.

So, you see why we might be interested in imposing inequality constraints and optimization problems from a data science viewpoint. A more sophisticated version of this idea is what is used in one of the data science algorithms called support vector machine. Though we will not study that technique in this first course on data science for engineers, I just wanted to point out that this class of optimization problems is very important from a data science viewpoint.

Now, let us go back to the same example that we had before, where we had the equality constraint and then we tried to solve the optimization problem and then just make that equality into an inequality. So, in this case let us assume what was equal to 12 in the last lecture has now become less than equal to 12 and let us understand intuitively what happens to problems this of this type. Now in the previous case we said when we have an equality constraint we said we are interested in any point on this line as a candidate solution these are all called the feasible points and of all of these points we were trying to pick the point which will give me the minimum objective function value.

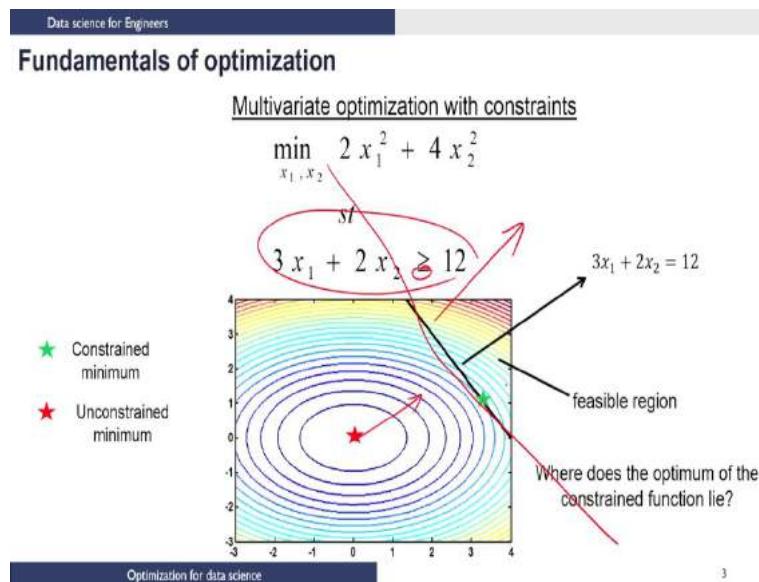
Now, when I have the optimization problem where I have this inequality and let us assume this is less than equal to in this picture what we have plotted is that the original unconstrained optimum or the unconstrained minimum is still the red star. Now if you look at this and then say this less than equal to 12 it can be basically rewritten as $3x_1 + 2x_2 - 12 \leq 0$ and remember that this is of the form in transpose $x + b \leq 0$. So, it is going to be one half space depending on how n is defined. In this case it will turn out that this is the half space that is represented by this equation.

So, the difference between the equality constraint and the inequality constraint is the following, in the equality constraint we had every point on this line being a feasible solution when you make this as a

inequality constraint. Then what happens is if you think of this line is extended all the way, any point to this half space now becomes a feasible solution and because every point in this half space is a feasible solution the unconstrained minimum also becomes a feasible point so the constrained minimum and unconstrained minimum are the same so this is an interesting thing to see.

So we saw that in the case of equality the unconstrained and constrained minimum were different, but when we made this into an inequality with the less than equal to sign we see that the constrained and unconstrained minimum are the same points.

(Refer Slide Time: 05:38)



Now, let us try and see what happens if I flip the sign and then said this is greater than equal to 12. Then what would happen is if you were to extend this line all the way, then the feasible region is to this side. Now, you asked the question where does the optimum lie? You will notice that again we know the best solution is here and as we move away from this solution, we will see that we are losing out on the objective function value and as before we know any point on this line or to the side is a feasible point and any point on this line is also feasible because of this equality sign.

Now, making the same arguments that we made in the case of the equality constraint problem, we will see that I give up on my optimality, that is I keep going through contours of larger and larger size where the optimum value keeps increasing and when a contour particular contour touches this line exactly at one point then I have a feasible point which is going to satisfy this constraint, the equality part

of the constraint. So, it is satisfying the general constraint and that is the worst I have lost in terms of how much my objective function has increased its value by.

Anything more would be unnecessary because if I move little further I am going to make my objective function value worse. However, there is no need to do that because I already found a feasible point here. So, this case the constrained minimum becomes the same as the minimum that we achieved with the equality constraint case. So, you see that depending on what type of inequality these different things can happen.

(Refer Slide Time: 07:46)

Multivariate optimization

$$\min_{\bar{x}} \quad f(\bar{x})$$

$$\text{st} \quad \begin{aligned} h_i(\bar{x}) &= 0, i = 1, \dots, m \\ g_j(\bar{x}) &\leq 0, j = 1, 2, \dots, l \end{aligned}$$

$g_j(x) \geq 0$

$-g_j(x) \leq 0$

Necessary condition for \bar{x}^* to be the minimizer

KKT conditions has to be satisfied

Sufficient condition

$\nabla^2 L(\bar{x}^*)$ has to be positive definite

Optimization for data science

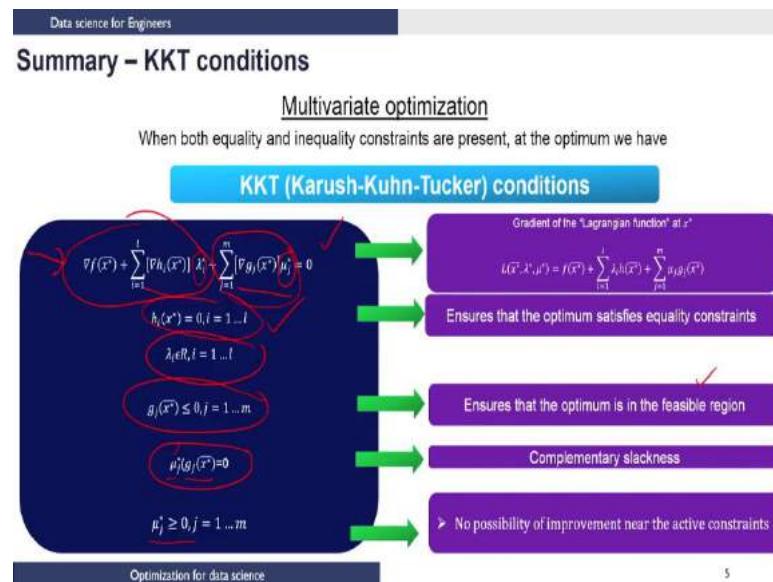
Now, from a general mathematical formulation viewpoint, we are going to generalize this. What I am going to show is I am going to show you the general conditions. Now for this course we might not be using this later in any data science algorithm. Nonetheless this forms a basic idea for more sophisticated data science algorithms like SVM and so on. So, it is worthwhile to pay attention and then understand this completely. So, general multivariate optimization problem we can say has both equality and inequality constraints.

So, I have the objective function, I have m equality constraints and l inequality constraints. Notice that we can always write the equality condition in this form where I have 0 on the right hand side because if you have something on the right hand side I simply move it to the left hand side and get a 0 on the right hand side. Similarly, any condition inequality condition whether it is greater than equal to 0 or less than equal to 0 I can always put it in this form. If the original condition is already in less than equal to form then it is the same as this if it is in the

greater than equal to form then what I do is I multiply by a negative and then I have this condition has less than equal to 0.

So, now, I call this my constraint so I can again put it in this form. Now this becomes a little more complicated formulation and I am going to show you the conditions for this in the next slide. Which will look a little complicated in terms of all the math that is there. What I am going to do is I am just going to simply read out the conditions in the next slide and then we will take a particular example and then demonstrate how the conditions work.

(Refer Slide Time: 09:45)



These are the conditions for multivariate optimization problem with both equality and inequality constraints to be at it is optimum value and let us look at this carefully. Remember when there was only equality constraints, we had this part of the expression where we had ∇ of f + linear sum of λ times ∇h for each one of the equality constraints.

Now, when you also have inequality constraints, what happens to this equation is, just like how we added a single parameter for every equality constraint, we add a parameter for each inequality constraints. So, if there are m inequality constraints there will be m parameters and in typical optimization books and literature people use λ as a scalar multiple for each one of these constraints. So, if there are 1 equality constraints, there will be 1 lambdas and people use the nomenclature of μ for the inequality constraints. So, if we have m μ 's there will be, there will be m μ 's corresponding to the m inequality constraints.

So, the difference between this condition in the equality and inequality case

is that for every one of these inequality constraints you add more linear combinations of μ times δg . So, look at this, this is the same form of as this except that I used λ here and μ here, but I take a take a ∇ of h and ∇ of g . That is the first set of conditions then much like how we had the constraints equality constraints also as part of conditions in the previous case, I am going to have the optimum solution satisfy all of this equality constraints.

Now the λ is some real number, so as many real numbers as there are equality constraints and much like how I still need to have this equality condition satisfied the optimum point, I need to have the inequality constraint also to be satisfied by the optimum point. So this ensures that the optimum point is in the feasible region. Now this real differences between the equality constraint condition and the inequality constrained situation shows up. We also have additional constraints, which are of this form, these are called complementary slackness condition.

So, what this says is if you take a product of the inequality constraint and the corresponding μ_j then that has to be 0. Basically what it means is either μ_j is 0 in which case this is free to be any value such that this condition is satisfied or this is 0 in which case I have to compute a μ and the μ that I compute has to be such that it is a positive number or it is greater than equal to 0. So, this condition is there to ensure that whatever optimum point that you have, there is no possibility of improvement -any more improvement- from the optimum point so that is the reason why this condition is there.

Now just keep in mind that if you are seeing course on optimization for the first time it is not very easy or natural to understand this constraints right away. However, what we are going to do is in the next slide we will take an example and then show you how these things work. One thing that I want you to keep in mind is if we had let us say an unconstrained optimization problem objective function in n variables, I always look at whether the optimum conditions have enough equations and variables for me to be able to solve the system of equations and clearly you know in the unconstrained case you have n equations and n variables and I clearly made the point in the equality constraint case that for every equality constraint you add an extra parameter.

However, you will have enough equations and variables because when you write the first condition which is of this form you will get the n equations and you will get as many equality constraints that need to be satisfied as there are lambdas. So that also works out properly. Now we will see whether the same thing happens in the case of inequality constraints.

(Refer Slide Time: 15:02)

Summary – KKT conditionsMultivariate optimization

- In general it is difficult to use the KKT conditions to solve for the optimum of an inequality constrained problem (than for a problem with equality constraints only) because we do not know *a priori* which constraints are active at the optimum.
- Makes this a combinatorial problem ✓
- KKT conditions are used to verify that a point we have reached is a candidate optimal solution.
- Given a point, it is easy to check which constraints are binding.



Optimization for data science

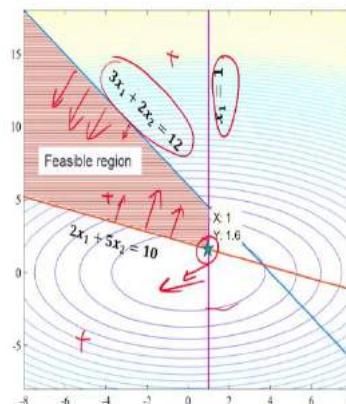
There is a specific problem in solving this -what are called the KKT conditions-the conditions that I showed in the previous slide are called the KKT conditions. It is not easy to solve the KKT conditions directly in the inequality case because of the complimentary slackness condition which says either μ could be 0 or z could be 0. So, we have to make a choice as to which is 0 so that makes this a combinatorial problem.

So, in general the KKT conditions are generally used to verify if a solution that we have is an optimal solution. However, optimization algorithms will have different ways of solving this problem and there are methods called penalty methods, active set methods and so on, we are not going to talk about those methods here. Nonetheless I just want you to understand how the solution works out in an analytical case.

(Refer Slide Time: 16:07)

Fundamentals of optimizationMultivariate optimization-quadratic programming

$$\begin{aligned} \min_{x_1, x_2} & 2x_1^2 + 4x_2^2 \\ \text{s.t.} & 3x_1 + 2x_2 \leq 12 \\ & 2x_1 + 5x_2 \geq 10 \\ & x_1 \leq 1 \end{aligned}$$



Optimization for data science

Let us take a look at a numerical example to bring together all the ideas that we have described till now. In this particular case it is a multivariate optimization problem which is actually called quadratic programming and this is called quadratic programming because the objective function is quadratic and the constraints are linear. Those types of problems are called quadratic programming problems.

I think this is the same objective that we have been using till now in the several examples. So, let us say this is the objective function and let us assume that we have constraints of the form shown here. Let us assume the first constraint is $3x_1 + 2x_2 \leq 12$, the second constraint is $2x_1 + 5x_2 \geq 10$ and the third constraint is $x_1 \leq 1$. Now I am not doing anything more to this problem, but nonetheless I just want you to remember that to be consistent with whatever we have been saying this should be converted to a less than equal to constraint which we will see how that happens in the next slide.

Now let us look at this pictorially. Remember that the value of the objective function is going to be plotted in the z direction coming out of the plane of the screen that you are seeing. So, the representation of that are these contours that we see here, which are constant objective function contours. So, I am just trying to see and explain how this picture speaks to both the objective function and the constraints. So, the objective function is actually represented in this picture as this constant value contours. So, if I am moving on this the objective function value the same we have repeated this several times.

Now in this $x_1 x_2$ plane, let us look at how these constraints look. So, I have this equation which is the equation of a line. So, when we talk about the first constraint which is less than equal to 12, then any point to this side of the line is a feasible point. Now, when we look at this constraint here then the equation of the line is $2x_1 + 5x_2 = 10$ and whenever I have something greater than = 10, this region is a feasible region and this is the $x_1 = 1$ line and $x_1 \leq 1$ would be this region.

So, if you put all of these regions together the only region which is feasible is shaded in brown colour here. So, if you take a point any point in this region it will be satisfying this constraint because it is to this side of this line, it will satisfy the second constraint because it is to the side of the line and it will satisfy the third constraint because it is to this side of the line. Notice that if you take any point anywhere else you will not be feasible. For example, a point here would violate constraint 1, but it would be feasible from constraint 2 and 3 viewpoint nonetheless all the constraints have to be satisfied. Now similarly if you take a point here while it satisfies constraint 1 and 3 it will violate constraint 2.

Now, if you notice the optimum point is going to lie here and we are going to try and find out this value through the conditions that we described in the last few slides.

(Refer Slide Time: 20:10)

Data science for Engineers

Fundamentals of optimization

Multivariate optimization-quadratic programming

$\min_{x_1, x_2} 2x_1^2 + 4x_2^2$
st
 $3x_1 + 2x_2 \leq 12 \Rightarrow (a)$
 $2x_1 + 5x_2 \geq 10 \Rightarrow (b)$
 $x_1 \leq 1 \Rightarrow (c)$

- Lagrangian

$$L(x_1, x_2, \mu_1, \mu_2, \mu_3) = 2x_1^2 + 4x_2^2 + \mu_1(3x_1 + 2x_2 - 12) + \mu_2(10 - 2x_1 - 5x_2) + \mu_3(x_1 - 1)$$
- First order KKT conditions

$$\begin{aligned} 4x_1 + 3\mu_1 - 2\mu_2 + \mu_3 &= 0 \\ 8x_2 + 2\mu_1 - 5\mu_2 &= 0 \\ \mu_1(3x_1 + 2x_2 - 12) &= 0 \\ \mu_2(10 - 2x_1 - 5x_2) &= 0 \\ \mu_3(x_1 - 1) &= 0 \\ \mu_i &\geq 0 \end{aligned}$$

So, what we do here is the following. So, we define something called a Lagrangian, which basically will boil down to the kind of conditions that we have been talking about I will explain this quickly. So, we take the objective function value or the objective function expression here and then there are no equality constraints here, there are only inequality constraints. I said we had one parameter age for each of these inequality constraints.

So, this is already in a form that is palatable to us which is less than equal to form. So, I just simply write this as μ one times $3x_1 + 2x_2 - 12 \leq 0$. So, this is already in the less than = form, when I look at the second equation, I want to make it into a less than = form then what I do is I said $10 - 2x_1 - 5x_2 \leq 0$. So, if I put this here this is into the less than = form and then corresponding to this if I have μ_1 corresponding to this μ_2 and corresponding to this I have μ_3 I have $x_1 - 1 \leq 0$ so you see that term here.

Then what you do is, you differentiate this with respect to x_1 and x_2 . You will get the first 2 conditions that we have been talking about for the constrained case with equality constraint, unconstrained case and so on. So, the 2 equations for the 2 decision variables so, when you differentiate this whole expression with respect to x_1 4 x_1 comes out of this term right here and this is only a function of x_2 . So, differential with respect to x_1 will become 0 and I will have $3\mu_1 + 1$ when I differentiate this with respect to x_1 I get $3\mu_1$ and from the second term I will get $-2\mu_2$ and from the third term I will get μ_3 . So, $\text{th}=0$. So

basically this equation you have and then when I differentiate the same expression with respect to x_2 .

So, I get $8x_2$ from this here and from the second term I get $2\mu_1$, I get $-5\mu_2$ from this term and from this term I get nothing because it is only a function of x_1 . So, I get this equal to 0 so, this basically gives you the condition that that we have talked about which is of the form in the equality constraint case remember I said you have $-\nabla$ has to be $= \sigma \lambda I \nabla$ of h_i and in the inequality case you also add $+\sigma \mu_i \nabla$ of g_i if g_i are the inequality constraints.

So, you kind of back out these 2 equations from this constraint and you have 2 equations here because ∇ of x is of size 2 by 1 h is 2 by 1 2 by 1 because there are 2 decision variables and these are scalars and this is a linear weighted sum. So, if you notice all of this, you see that I have 2 equations. However, I have 5 variables that I need to compute, I need to compute a value for x_1 , I need to compute a value for x_2 and then I need to compute μ_1 , μ_2 and μ_3 . So, let us see how we do that. We go back and add the complementary slackness conditions. Also keep in mind that other than this we also have to make sure that whatever solution we get still has to satisfy the 2 inequality constraints also.

So, whatever solution we get should still be such that $2x_3 + 2x_2 - 12$ is less than $= 0$ and $10 - 2x_1 - 5x_2$ is less than $= 0$. So, these 2 also have to be valid, but because these are inequality constraints we cannot actually use them to solve for anything. Once we solve for these variables we have to still verify whether these conditions are satisfied or not. So, going back to the complementary slackness condition we saw that we will have μ times g is 0. So, this is corresponding to the first inequality constraint, this is corresponding to the second inequality constraint and this is corresponding to the third inequality constraints and we also have to have this μ_i greater than $= 0$. So, all of these conditions have to be satisfied for an optimum point.

Now, it is interesting to notice something here for let us just take this as an example already we have 2 equations, I have already mentioned that the 2 equations are here. So, I am looking for another 3 equations to compute μ_1 , μ_2 , and μ_3 . So, that will be a total of 5 equations and 5 variables. So, for this to be 0 you could say in this equation either μ_1 is 0 or whatever is inside the bracket is 0. So, we could say one possibility is whatever is inside the bracket is not 0 in which case μ_1 has to be 0. So, the key point that I want to make here is if we say whatever is in the bracket is not 0, then that automatically gives me the value for μ_1 . So, out of the 5 variables here I have already computed one, similarly if I say whatever is inside here is not 0 then μ_2 has to be 0 and whatever is inside here is not 0 μ_3 has to be 0 then I have already computed values for μ_1 , μ_2 , μ_3 .

Then I could substitute those values back into this equation and I have 2 equations I can calculate x_1 and x_2 . So, this is one option. So, in one of the previous slides I had mentioned that this becomes a combinatorial problem because we could also assume that this goes to 0. So, this term goes to 0 and now let us assume actually this is nonzero, this is nonzero, let us see what happens to that case do we have enough equations and variables.

So, in this case we will have 1 equation, 2 equations, the third equation will be μ_2 has to be 0 because we have assumed this is not 0 the fourth equation has to be μ_3 is 0 because we have assumed this is not 0. Now the fth equation becomes the one which we have assumed which is the term inside the bracket is 0. So, in which case again I have 1 2 3 and then $\mu_2 = 0$ $\mu_3 = 0$ as 5 equations in 5 variables. You could for example, assume that this and this are 0 in which case I have to compute μ_2 and μ_1 and then let us say this is not 0 then μ_3 is 0, but in that case also you will have 1 equation, 2 equation $\text{th}=0$ is 1 equation and $\text{th}=0$ is 1 equation. So, again I have 5 equations and 5 variables

So, whichever assumption you make as far as these equations are concerned you will have enough equations and enough variables. However, for some of those choices when you actually find a solution and try to plug this back into this right here it might not satisfy this. So, in which case that is not a viable option for us from an optimization viewpoint. And in some cases this might be satisfied, but the μ that you calculate out of the equations you get might not be positive. So, it might get negative μ in which case again this is not an optimum solution. So, let us see how this happens for this example.

Now, I am going to use one notation here so, that we can understand the table in the next line. So, whenever I assume that an equality constraint is exactly satisfied; that means, when I say $3x_1 + 2x_2 - 12 = 0$, then we say this constraint is active it is active because the point is already on the constraint. If I take a point here then that is not on the line so, that is basically less than $= 0$ so, I will say it is inactive. So, for every constraint I can say whether it is active or inactive. So, if this constraint is active; that means, $3x_1 + 2x_2 - 12 = 0$. If this constraint is active; that means, $2x_1 + 5x_2 - 10 = 0$ and if this constraint is active; that means, $x_1 = 1$.

(Refer Slide Time: 29:38)

Fundamentals of optimization

Multivariate optimization-quadratic programming

Sl.no	Active (A) /inactive (I) constraints			Solution (x, μ)	Possible optima (Y/N)	Remark
	(a)	(b)	(c)			
1	A	A	A	Infeasible	N	Equations do not have a valid solution.
2	A	A	I	$x = [3.6364 \quad 0.5455]$ $\mu = [-5.2 \quad -1.45 \quad 0]$	N	$x_1 \leq 1$ is not satisfied, $\mu_1 < 0, \mu_2 < 0$
3	A	I	A	$x = [1 \quad 4.5]$ $\mu = [-18 \quad 0 \quad 50]$	N	$\mu_1 < 0$
4	I	A	A	$x = [1 \quad 1.6]$ $\mu = [0 \quad 2.56 \quad 1.12]$	Y	All constraints and KKT conditions satisfied ✓
5	A	I	I	$x = [3.27 \quad 1.09]$ $\mu = [-4.38 \quad 0 \quad 0]$	N	$x_1 \leq 1$ is not satisfied
6	I	A	I	$x \in [1.21 \quad 1.51]$ $\mu = [0 \quad 2.45 \quad 0]$	N	$x_1 \leq 1$ is not satisfied
7	I	I	A	$x = [1 \quad 0]$ $\mu = [0 \quad 0 \quad -4]$	N	$2x_1 + 5x_2 \geq 10$ is not satisfied
8	I	I	I	$x = [0 \quad 0]$ $\mu = [0 \quad 0 \quad 0]$	N	$2x_1 + 5x_2 \geq 10$ is not satisfied

So, in the example that we are considering right now, there are 3 inequality constraints and as I mentioned in the previous slide if the inequality constraint is exactly satisfied that does it becomes an equality constraint then we call that an active constraint and if the constraint is not exactly satisfied we call it as an inactive constraint. And now we have 3 inequality constraints and each of these constraints could be either active or inactive.

So, there are 2 possibilities for each of these constraints and since we have 3 constraints there are $2^3 = 8$ possibilities which = the 8 possibilities that we have here. So, what I am going to do in this case is we are going to enumerate all possibilities for you to get a good understanding of how this approach works when you have inequality constraints. So, let me pick let us say a couple of rows from this table to explain the ideas behind how this works and what we are going to do is in the next slide we are actually going to see graphically what each of this case means.

So, let us look at the first row for example, here the choice we have made is all the 3 inequality constraints are active. That means, they all become equality constraints. Notice something interesting remember there are 2 decision variables. So, each inequality constraint is basically representing one half space for a line and when they become active each of these constraints become an equality constraint each of them become line.

Now when all 3 are active then we have 3 equations in 2 dimensions right. So, there are only decision variables x_1 and x_2 , but I have 3 equations in those 2 variables and from our linear algebra lectures we know that when we have more equations than variables in many cases we are not going to find a solution for the 2 variables which will satisfy

all the 3 equations. So, in this case you cannot solve this problem it is infeasible because though I have enough equations a subset of these equations 3 equations are in 2 variables and I cannot find a valid solution. We will understand what this is geometrically in the next slide.

Let us look at some other condition here. Let us pick for example, row 5. So, if you look at row 5, we have made a choice that the first constraint is active, the second constraint is inactive and the third constraint is inactive, that basically means the first constraint equal to 0, the second and third constraints have to be less than equal to 0, which needs to be tested after we go through the solution process. Now much like how I described before in this case also we will be able to find 5 equations and 5 variables and we can solve for x_1 and x_2 which is shown here and we have solved for μ which is shown here.

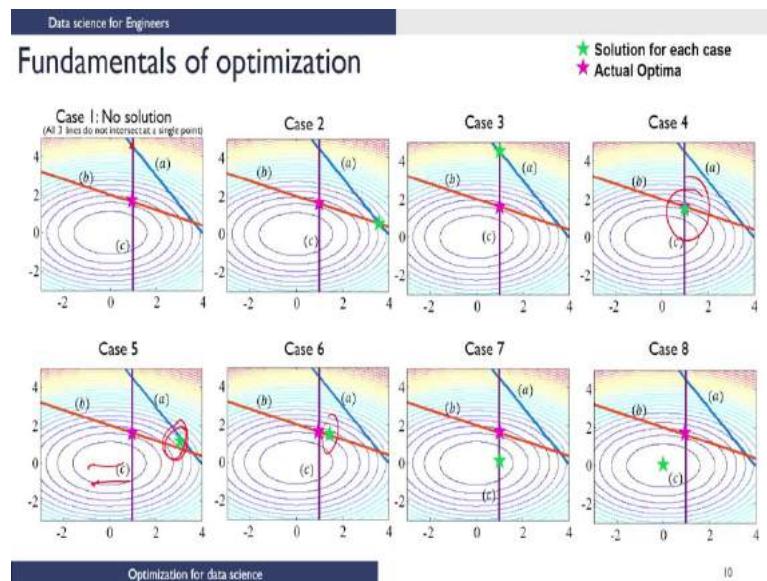
Now if you just look at this solution, you will say it seems to satisfy everything because I have a solution for x_1 and x_2 and I have μ_1, μ_2, μ_3 where μ_1 is - 4.36 and μ_2 and μ_3 are 0. However, when you look at this μ you will see that one of the μ s is negative. Which basically means that this cannot be an optimum point based on the conditions we showed in a couple of slides back, not only that on top of it when you actually put these 2 values into the constraint x_1 is less than equal to 1 it is not satisfied because x_1 is 3.27.

So, if you take this row for example, it looks like both this μ being positive is not satisfied and this constraint is also not satisfied. An interesting thing to note here is we have to go back and check only the constraints that we have assumed to be inactive because the active constraint is already at 0. So, whatever solution again will automatically satisfy the active constraint. Let us say row 6 for example, here we have made a choice that the first constraint is inactive, the second constraint is active and the third constraint is inactive.

Again we will get 5 equations and 5 variables and we will look at the solution here. It again looks good from the x view point. We have solved for x_1 and x_2 , 1.21 and 1.51. Unlike the previous case in this case the μ s also looked good because μ s have to be positive. So, I have 0 to 0.450. So, it looks like this, this is a good candidate for an optimum point. But once you have satisfied all of this you have to still go back and look at whether the inactive constraints are satisfied by this solution point right here and the inactive constraint here is x_1 has to be less than equal to 1, but if you notice that 1.21 does not satisfy this constraint. So, this cannot be an optimum either.

When you look at row 4, where we have assumed constraint 1 is inactive and 2 and 3 are active, I get a solution $x_1 = 1.6$. I get all μ 's to be positive which is also one of the conditions for an optimum point and then I have to only verify this constraint here because other 2 are active constraints and they are providing equations for us to solve so they are like they are going to be valid. Now when you put these 2 values into the first constraint you will check that it actually satisfies the inequality also. So, this is a case where all constraints and KKT conditions are satisfied. So, this is the optimum point so, the optimum solution is 1.6 which is what we had indicated in the slide with the geometry of this problem.

(Refer Slide Time: 36:56)



So, let us look at each of this case in terms of where the optimum lies and how we interpret this. So, if you take the case one where we took all the constraints to be active we said we have 2 variables and 3 equations and that is not satisfiable in general. Geometrically what this means is we want the point to lie on all the 3 lines at the same time because we have assumed all 3 lines are active concerned. That means, all of them have to be equal to 0.

So, you find that you cannot get a point where the all 3 lines equations are satisfied. So, if I want to satisfy 2, I will get a point here nonetheless it would not satisfy the other constraint and so, on. So, this is the case of not all 3 lines intersect so we do not have a solution here. Similarly you can look at each of these cases and you can see what happens in each of these and you will notice that only in case 4 will you have the solution that you got from the Kuhn Tucker condition and the actual optimum being satisfied.

In every other case you will see that there is some problem or other. So, if you go back I think we looked at case 5 and 6 then the solution for case 5 is this, the solution for case 6 is this and we said these two are not good solutions because they violate the condition x_1 is less than equal to 1 which basically seen here because the feasible region is to this side of line x_1 is less than equal 1, but this point is violating this constraint and here again you see that this point is violating this constraint.

So, in summary what you need to know is that in the unconstrained case it is very clear that I just simply write this ∇ of f is 0 as the condition and I will have enough equations and variables there will be n equations and n variables. In the constraint case with equality constraints I will get the same n equations, but the form will be slightly different we write that as $-\nabla f$ is $\sigma \lambda_i \nabla$ of h_i and since we add as many variables as there are equality constraints and since the equality constraints have to be satisfied those give you the extra equation. So, you will have enough equations and variables.

In the inequality constraint case, the first n equations come from a very similar form where $-\nabla f$ is $\sigma \lambda_i \nabla$ of $h_i + \sigma \mu_i \nabla$ of g_i that gives you n equations and corresponding to every one of those λ corresponding to equality constraints you will get so many equations which are the equalities have to be satisfied. The only complication comes in when you have these inequality constraints where either you know you have can have one or the other be 0 that is what we call as complementary slackness.

So, we have this μ times g going to be 0. So, if g is 0 we call that as an active constraint in which case we have to calculate the μ corresponding to it and in the optimum point μ will be greater than equal to 0. Now depending on the form in which you write whether you write all of these constraints or as inequality constraints less than equal to or greater than equal to and also in terms of whether you write the original equation as $-\nabla f$ equal to this sum on the right hand side or ∇ of equal to sum on the right hand side the sine of μ in different textbooks and different papers might be reported as either they have to be positive or they have to be negative.

So, you have to be careful about the conditions when you look at those, but if you stick to this type of writing the equations, where you write $-\nabla$ of a $\sigma \lambda \nabla h_i + \sigma \mu \nabla g_i$ and if you write all the constraints as less than equal to inequality constraints, then μ 's have to be positive for the point to be an optimum point which is what we saw here.

As I mentioned before while this unconstrained case is a very very important case in machine learning algorithms such as SVM and so on the constrained case I mean we might not be using this quite a bit in this course. Nonetheless for the sake of completeness and for the sake

of giving the foundations for understanding other data science algorithms that outside of this course that you might go and study.

I have also described the key ideas behind how to solve constrained optimization problems when you have equality and inequality constraints. Keep in mind that while I have shown you the conditions, I have not shown a proof or a derivation of these conditions in a formal manner. The equality constraint case I appealed to intuition to tell you why the conditions turn out to be the way they are. However, all of this can be formally proved and you can derive these conditions based on formal mathematical arguments.

So, with this we conclude this portion on optimization for data science now we have all the tools that we need to understand data science problems. The next set of lectures what I am going to do is I am going to introduce to you different types of data science problems that we encounter, how do we think about these data science problems. Is there some formal way of thinking about these problems; that we can use to solve a variety of problems and then we will move on to regression as a function approximation tool and we will look at different clustering techniques that are used in data science and then we will finally conclude with one particular technique called principle component analysis which is very useful for engineers and end with more general example of how one solves real life data science problems.

So, I hope to see you continue these lectures and understand more of data science.

Thank you.

Data Science for Engineers
Prof. Ragunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 29
Introduction to Data Science

Now, that we have looked at the fundamentals required to understand data science in terms of linear algebra statistics and optimization, we are going to start series of lectures where we introduce data science, describe different techniques that are used in data science and finally, end with one practical industrial example of use of data science. While we introduce the techniques we will also use smaller examples to illustrate how the technique might be used in a data science problem. At the end of this course there will also be a case study for the participants to practice.

So, this is the first lecture on introduction to data science. Before we jump into the techniques I would like to introduce some interesting ways of looking at data science and in a broader context understand what these techniques are doing and how one should think about data science problems. One could teach these techniques as disparate set of methods to solve data science problems; however, the critical thing is in learning how to use these techniques for real problems which is what we call as problem formulation.

What we will do in this course is introduce the participants to a data science problem solving framework, very short lecture on that, to give you a view of how one should think about general data science problems and how you convert a problem you know which is not well defined into something that is manageable using the techniques such you learn in this course.

So, let me start with this laundry list of techniques that people usually see when they look at any curriculum for data science or any website which talks about data science or many books that talk about data science. I have just done some colour coding in terms of the techniques that you will see in this course in green.

(Refer Slide Time: 03:05)

Techniques

- Regression analysis
- K-nearest-neighbor
- K-means clustering
- Logistics regression
- Principal Component Analysis
- Predictive Modeling
 - Lasso, Elastic net

And other techniques are out there which we will not be teaching in this course, but which would be a part of more advanced course. So, there are techniques such as Regression analysis, K - nearest - neighbour, K - means clustering, logistics regression, Principle component analysis, all of which you will see in this course then people talk about Predictive modelling under that there are techniques such as Lasso, Elastic net that you can learn.

(Refer Slide Time: 03:38)

Topics

- Linear discriminant analysis (LDA)
- Support Vector Machines
- Decision trees and random forests
- Quadratic discriminant analysis (QDA)
- Naïve Bayes classifier
- Hierarchical clustering

**What types of problems are being solved ? Why
are there so many techniques?**

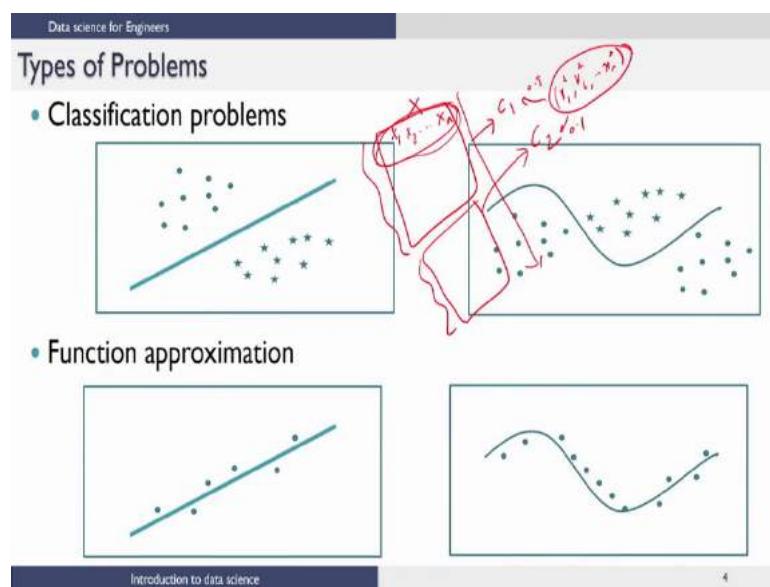
Then there are techniques such as Linear discriminant analysis, Support Vector Machines, Decision trees and random forests,

Quadratic discriminant analysis, Naive Bayes classifier, Hierarchical clustering and many more such as deep networks and so on. So, to get a general idea of data science one might be tempted to ask that if all of these collections of techniques solve data science problems then, one would like to know what types of problems are being solved really and once one understands the types of problems that are being solved then, the next logical question would be - do you need so many techniques for the types of problems that you are trying to solve.

So, this would be typical questions that that one might be interested in answering. What I am going to do is I am going to give you my view of the types of problems that are being solved and why there are so many techniques to solve these types of problems. Since this is a first course on data science for engineers we going to cover major categories of problems that are of most interest to engineers. This is not to say that other categories of problems do not exist or that they are not interesting.

We will keep this viewpoint in the background as we go through the lecture materials of this course. Other than this one could also think of statistics as useful by itself for data science problems. Statistics is also intricately embedded into the data science techniques in terms of the formulation themselves and also in characterizing the properties of the machine learning techniques.

(Refer Slide Time: 05:39)



So, in my mind fundamentally I would say that there are mainly 2 class of problems that we solve in data science. So, I am going to call these as classification problems and function approximation problems.

So, let us look at what classification problems relate to so these are types of problems where you have data which are in general labeled and I will explain what label means and whenever you get a new data you want to assign a label to that data.

So, typical example of this type of problem is called a binary classification problem which is used in many applications I will point out 2 applications for example. In this type of problem what you have is data we will call data x . This data could have many attributes let us say x_1, x_2 all the way up to x_n this is something that we saw in linear algebra and so on and in binary classification problems what you have is you have a group of data which you say can be assigned a label let us say c_1 and I will explain why I use the term c , c refers to the class to which this data belongs and then another block of data with the same attributes may be labeled as c_2 .

So, now the data science problem is the following if I give you a new data point let us say x_1, x_2, \dots, x_n , the algorithm should be able to classify and say this point is likely to have come from either class 1 or it could have come from class 2. So, assigning a label to this new data in terms of what is the likelihood of this data having come from either class 1 or class 2 is the classification problem. Let us say if you assign the likelihood of this coming from class 1 as 0.9 and from class 2 as 0.1 then one would make the judgment that this data point is likely to belong to class 1.

Now, let us see how this is useful in a real problem. So, I will give you 2 examples one example is something that people talk about all the time nowadays which is called fraud detection. So, let us take one particular case of fraud detection for example, so, whenever we go and use our credit card we buy something and the credit card gets charged. So, let us say there are certain characteristics of every transaction that you record such as the amount the time of the day the transaction is made the place from which the transaction is made the type of product that is bought through the transaction and so on. So, you can think of many such attributes. Let us say those are the attributes that characterize every single transaction.

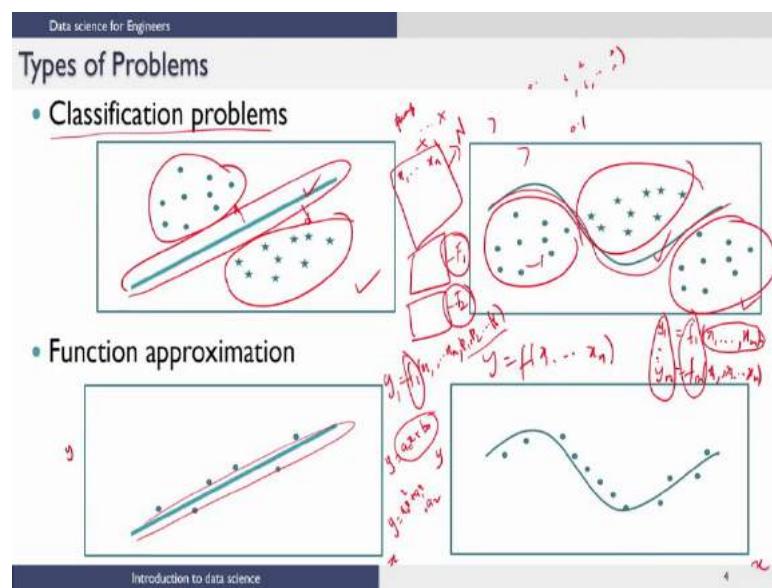
Let us assume that there are many people and they are making transactions and you have transactions listed like this and you find out that of these, these were actually fraudulent transactions these were transactions that was not legal or was not made by the person who owns the credit card and these are transactions which are legal. So, this is something that you label based on exploring each transaction which you think might not be right and actually when you find out that that transaction was not legal then you put it into the basket which is illegal transaction.

Now, if you use a data science algorithm a binary classification algorithm to be able to give the likelihood of a transaction being correct or fraudulent based on this easily calculatable attributes or easily monitorable or measurable attributes. Then, whenever a new transaction takes place you could run it through this classifier and then find the likelihood of this transaction being fraudulent.

And in cases where the transaction has a very high likelihood of being fraudulent, then the company could call that person and then say hey we saw that your credit card was used in such and such a place such and such a time for buying such and such a thing did you actually do this transaction and if you have gone on a vacation to remote place and you made this transaction you tell them I have come to this place on vacation this is the right transaction and so on if not, then you find that transaction is fraudulent and stop the payment. So, this is one example of how a binary classification problem is useful in real life.

Now, when we talked about this data and we talked about the binary classification problem, we talked about just 2 classes x_1 and x_2 , but in reality there could be problems where there are multiple classes. One very good engineering example would be fault diagnosis or prediction of failures. Where you might have, let us say a certain equipment a pump or a compressor or distillation column whatever the equipment might be and then the working of that equipment is let us say characterized by several attributes. How much power it draws, how much performance does it give, is there vibration, is there noise, what is the temperature and so on.

(Refer Slide Time: 13:12)



So, now, you could have engineering data x which let us say talks about the characteristic of let us say a pump and the pump is characterized, the operation of the pump is characterized, by let us say several attributes x_1 to x_n . And if you have legacy data or historical data where you have been operating pumps for years and years and then you know that if these variables take values in this block then everything is fine with the pump.

So, I write n for normal and then you could have a block of data and that data might have been the data that is recorded whenever there is a particular type of fault in the pump let me call this fault f one. Then you could have another block of data which could have been seen when there is fault f_2 and so on. So, we will just stick to 2 faults f_1 and f_2 . Let us assume these are the only 2 failure modes that are possible. Now you start operating the pump and then at some point you get this data and then you ask the following question. Based on this data would be possible for me to say if the pump is operating normally or is there likely to be failure mode 1 that is the current situation of the pump or is it failure mode 2 that is the current situation of the problem.

So, in this case you see that there are 3 classes n , f_1 and f_2 . So, this is what is called a multi class problem. So, again when a new data comes in we want to label this as either normal f_1 or f_2 if it is normal, you do not do anything. If it is f_1 , if it is very severe then you stop the pump and then x it. If it is not very severe you let the maintenance know that this pump is going to come up for maintenance at some time and in the next shutdown of the plant this pump needs to be maintained. So, that is how classification problems are very important in engineering context.

So, we will look at examples of both binary classification and multi class classification as we go through the series of lectures. So, in summary the one type of problem that we are interested in data science is classification and these 2 pictures here show the different types of challenges that we are going to face when we look at classification problems. So, problems where linear equation can be a decision function for us to classify are called linear classification problems or we call these problems are as linear classifiable or linearly classifiable and here we show in 2 dimensions, so, binary classification problem so all of this could be a class 1 and all of these could be class 2 and a line or a plane or a hyper plane could be used to classify this data points.

Now, more complicated problems are where hyper plane or a line might not be enough for us to classify. Here is an example of a classification problem which is non-linear. So, let us assume that this data and this data belongs to class 1 and this data belongs to class 2. However you try to draw a line it would be very difficult ,almost impossible in this case, to classify these 2 classes with just a line in this 2 D picture. However, if your decision boundary are the function that

you are going to use to classify is of this form. So, you see the difference between this and this, this is non-linear, this is linear, then we could easily extend the concepts that we have learnt in terms of the half spaces and so on to do classification for these types of problem using non-linear decision boundaries.

So, you would say if the points are to this side it is 1 class and if the points are to the other side it is another class. One has to do this carefully defining the equivalent ideas for non-linear decision boundaries equivalent to the linear case very carefully and the minute you move from linear to non-linear then there are a host of other questions that come about. And these questions are really related to what type of non-linear function should one use.

When we talk about linear classifiers the linear functional form is fixed it is very simple. It is only one functional form you have to estimate the parameters of course, but we do not have to really think about what functional form you were going to use. However, if it is a non-linear problem then we really need to choose a particular type of decision function that we need to use and how do you choose this decision functions now the minute you go to the non-linear domain there are infinite number of functional forms that you can choose how do you choose one that works for you it is an interesting and important question that one needs to answer.

So, that is as far as classification problems are concerned, now let us move on to the other type of problem that one solves in data science this is what I would call as function approximation problem. Again, I am showing function approximation problems in 2 dimensional space here. So, I might have an out-put and an input so again in a general case we will have many inputs and many outputs. This is what is called as a case of single input here and a single output. However, you could have many attributes and the output being a function of many attributes.

This is also a function approximation problem or you could also have many outputs which are a function of many attributes. So, this is also possible. So function approximation is the task of nding these functions and whenever we write a function this function is typically parameterized by parameter. So, for example, if you just take let us say one output and then say this is f_1 , x_1 , x_2 , x_n these are the attribute values and there will also usually be a set of parameters that you have to use for that function. So, that could be p_1 p_2 and p_r let us say.

So, when I talk about a function approximation problem, then the problem that we are trying to solve is the following. Given several samples of these out-puts and the corresponding attributes that resulted in these outputs. So, this is the data that we are going to talk about and once I have a large amount of this data, how do I find this function

form and once I choose a functional form how do I also identify the parameters that are in the functional form. So, a simple example is if it is a linear functional form then I say $y = a_0 x + b_0$ let us say.

In this case the functional form is linear and the parameters are a_0 and b naught if you assume that it is a quadratic functional form then you could do $a_0 x^2 + a_1 x + a_2$. So, in this case the functional form is quadratic and there are 3 parameters now a_0 , a_1 and a_2 . So, when you do this function approximation you will have to figure out both the function and the parameters and in classification problems you want to come up let us say in the linear case with a line or a hyper plane where these points are as far away from this as possible. In the function approximation case what you want to do is, you want to find a line or a hyper plane such that these points are clustered around that and this is a linear problem which is what we are going to see in this course as linear regression.

Now the same non-linear version of the problem similar to the picture on the top is shown here. Here you want to have a non-linear surface or a curve that goes through these points and these points are clustered around that curve. So, in summary there are really only two types of problems that we predominantly solve from an engineering viewpoint using data sciences, these are classification problems and function approximation problems.

So, if there are only two types of problems that we are really solving then one might ask why are there so many techniques for solving these types of problems and one standard question that comes about whenever someone does data science is if a particular technique is better than another technique and the proponent of one technique will say this is a greatest technique the proponent of other technique will say that is a greatest technique and you know this debate keeps going on and so on.

So, I am going to give a slightly different view of why we have so many techniques and you know you can kind of resolve in some sense this question of which technique is better. So to do this let us do a thought experiment.

(Refer Slide Time: 24:44)

Thought Experiment

- How many articles are in the table?



- We can count all that is there to see

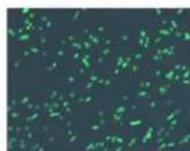


So, we have many objects on the table that is shown in the slide, then if I asked you how many articles are there in the table you would quickly say well there is a camera, there is a cup, there are two mobile phones, there is a watch, there is a pen, bottle and so on. So, basically we can kind of count or see whatever there is to see and then enumerate and then say these are the objects or articles in the table. So, in some sense we can count all that is there to see. So, this at this point you will say this is all that is on the table then I asked you the question is that all really that is there on the table.

(Refer Slide Time: 25:41)

Thought Experiment (Metaphorical)

- What about things that we cannot see?



- How do we understand things that we cannot see – appropriate fluorescence chemical?



And not to take this very literally, but this illustrates the key idea that I want to use when we go back to answering the question as to

why there are so many techniques for data science. So, carrying on I will ask you is that all there is on the table and then if I ask you this question what about things that we cannot really see. So, in the table there might be millions of teaming microorganisms which are not visible to us to the naked eye.

So, if I ask you to enumerate everything that is there on the table you can only enumerate what you can see the things that you cannot see you cannot enumerate. So, let us assume again you have to understand the logic behind what I am trying to explain not to take this too literally. Let us assume that you suspect there could be four different types of microorganisms also that could be on the table, now you cannot see it. So, what you do is just again to do the thought experiment let us say someone came up with some chemical which if you simply spray on you can start seeing these microorganisms.

So, let us say there are 4 microorganisms there are four chemicals, now the assumption here is when someone comes up with a chemical like this they have tested it, they have shown that it works for that particular microorganism very theoretically and repeatedly they have shown that it works. So, we cannot re-all go back and question whether this chemical is good for this microorganism because that has been demonstrated reasonably well.

So, if there are these 4 microorganisms what you would do is. You have to make an assumption as to what exists on the table. So, let us say you make the assumption that microorganism one is what is there on the table. So, you pick up the fluorescent chemical one and then spray it. Now if you see fluorescence and then you would come to the conclusion yes my assumption is right this is the microorganism that is also on the table.

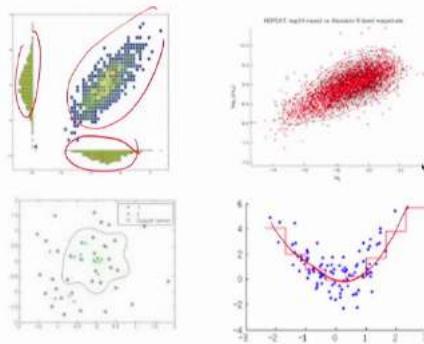
Now, the interesting thing is if it does not for us then the conclusion is not that the chemical is bad because that is been provably shown to work for this particular case, you would only assume that the assumption that you made is not right. So, you have to go back to the next assumption which would be microorganism 2 is there on the table and then you look at the fluorescence chemical 2 and so on.

So, once you do this exercise and let us say when you use chemical 2 and 4 it fluoresced one and 3 did not fluoresce then at the end of that exercise you could go back and then say the articles on the table or the camera and the and the cell phone and so on and also microorganism 2 and 4. Now notice how you have been able to see what you cannot visually see using this assumption validation cycle. So, this is an important thing to understand and I am going to connect this to the techniques and data science in the coming slide.

(Refer Slide Time: 29:34)

Thought Experiment

- If world were 2D?



- Data analytics not as critical

If world were 2 D for example, so all that we need to do could be done with just two attributes and looking at two attributes then whether it is a classification problem or a function approximation problem I can simply visualize it draw whatever I want characterize, and then be done with the problem. So, for example, here I could say this looks like a distribution; it looks like a normal distribution, in the two variables and so on.

If this were a function approximation problem you could really say well a single line will not solve this problem. So, maybe I can use a non-linear function and so on. So, if world were 2 D data analytics is still important because I need to explain the variability and all that; however, it is not as critical as the case where we have many more than two variables or more than two attributes and the situation currently is that for every problem that one tries to solve there is this data deluge.

There are tons of attributes that one could actually measure and monitor and you really want to see how many of these attributes are really going to contribute to the problem that we are trying to solve or how many of these attributes are really important. So, we are going we are going with big data from two dimensions to multiple, multiple dimensions and the question then is how do I understand the organization of the data, in multiple dimensions where I cannot see multiple dimensions I cannot see beyond 3 D.

(Refer Slide Time: 31:28)

Thought Experiment

- Data analytics tools are like a microscope to probe higher dimensional data



- Make assumptions that has the possibility of characterizing the higher dimensional data
 - Gaussian distribution
 - Linearly separable
 - Many more



Introduction to data science

So, how do I do this is a question that we are trying to answer. So, I would think of data analytic tools as microscope to probe higher dimensional data. Data in much more than 2 dimensions that you cannot actually see or visualize and the way we do this is the following. So, we have data and we cannot see it we cannot plot it because there are attributes in the 1000's- 10000's in some cases.

So, what you have to do is much like the microorganism example that I showed you. You have to make some assumptions about the data. To come up with a comprehensive set of assumptions is difficult, but let me explain some of the assumptions that are generally made. You could make the assumption that the data is actually random or data is generated from random process and you could make distribution assumptions such as it is Gaussian distribution and so on. Or you could make assumptions about how the data is organized, such as I think I can use a linear classifier to solve the classification problem in which case we are making the underlying assumption that the problem or the data is structured in such a way that it is linearly separable and there are many more assumptions that you can make it can make combinations of assumptions and so on.

So, you start with multi dimensional data you make these assumptions and then what you do is the following.

(Refer Slide Time: 33:17)

- questions about the data
- If the answers make sense then the data is "likely" to be organized in conformity with the assumptions
 - If the answers do not make sense, modify assumptions and choose (develop) a technique
 - Hopefully, the previous iteration can be analyzed carefully in the assumption modification process
 - Continue till the answers are satisfactory – Notice how we are seeing the "invisible"
 - Understand the importance of test data in the process
 - You now know why there are so many methods
 - Also tells you how you should choose a method

You pick a technique based on the assumptions and this technique should have been proven to solve problems where these assumptions have been made. So, in other words let us say if you make the assumption that the problem is linearly classifiable, then you really want to pick a technique which will work very well, which has been shown to theoretically work very well, for linearly separable problems.

So, this is equivalent to picking the chemical that has been shown to make a certain type of organism fluoresce. So, you choose the technique and then you deploy this technique and if the answer makes sense and we will see what it means when we say make sense mathematically from a data science viewpoint then the data is likely to be organized in conformity with the assumptions that you have made.

So, important the key, it is important to look at the key words that we are using it is "likely" to be organized and assumptions are important. So, likely would mean we will have to do some metric and different people will use different metrics and different levels of satisfaction of that metric to be convinced that what they have is right and wrong. So, that is where subjectivity comes in, but if the answers make sense then we will say the data is likely to be organized in conformity with the assumptions.

If the answers do not make sense, then typically the tendency is to blame the technique it is really not the technique that is a problem, the problem is the assumptions that we have made because we are not able to see this data in multiple dimensions. So, what you should do is you should modify the assumptions and choose our develop if you are a data scientist a technique to solve this problem if these assumptions were true.

Now, hopefully the previous iteration where you actually use some assumptions and saw that the assumptions were violated and that it was not likely that those assumptions are the one that are valid for this

problem. Even though you failed in that attempt you still got something out of it which would help you in modifying the assumption. So, this assumption modification process could be done with more knowledge from failed attempts from before.

Now you continue with this process till the answers are satisfactory and notice in this process how you are seeing the invisible. So, you are able to see data in n dimensions. So, for example, you cannot clearly see hundred variables plot them and then see whether they are linearly separable or not. But if you use a linear classifier and it worked very well then you know that the data is likely to be organized in such a way that a hyper plane could separate this data into two groups. So, you have started seeing the invisible much like the thought experiment we did with the table case.

Now, this question of likely and makes sense are very important. So, how do I ensure that I test to see whether the results that I have are good enough or not. That is done using test data in many of these data analytic techniques. So, the test data is very important when we do this exercise and as we teach different techniques you will see how this is important and will explain this in greater detail.

Now ultimately what I want to point out is the following now we have an answer for why there are so many methods. There are many types of assumptions that you could make and for each of these assumptions based on the assumptions you could come up with techniques which would work very well if those assumptions were true or the data was organized in a way the algorithm assumes it is organized.

So, since there are so many assumptions, there are many combinations of assumptions you can make. There are many techniques which are tuned and developed particularly to solve problems where data is organized according to the assumptions that are used in the technique. So, that is the reason why you have so many techniques. So, in some sense when you look at all of these techniques it is not as important or as interesting to compare these techniques blindly in terms of this is better than the other one and so on.

But it is more important from a data science perspective from a learning and understanding data science perspective to look at each technique in terms of the assumptions that it makes about the problem that is being solved and once you have a mental map of the assumptions that the technique uses to solve a problem and the technique, then you are in a good situation to be able to use a particular technique or a group of techniques for solving a particular problem so this is important to keep in mind.

So, in this lecture the first introductory lecture on data science I wanted to right away address the questions of the type of problems that we solved and in summary most of the problems that you solve in data science can be categorized as either classification problems or function approximation problems that is one take home message from this lecture and the other message is that there are several techniques for solving these data science problems we wanted to know why there are so many techniques.

So, I gave you a slightly different perspective on these techniques in terms of them allowing us to see or visualize or characterize or explained data in multiple dimensions. So, you start seeing data in multiple dimensions which is not possible otherwise. What we will do in the next lecture is to get some of these ideas into a notion of a framework for solving data science problems and I will illustrate that framework using one activity in data science problems which is used in many many problems this is in general the first step in many data science problems which is called data imputation.

So, I will describe a framework for solving data science problems and use this data imputation as an example to explain what that framework is and how does it work. And as part of that process, we will also see how we use this assumption validation cycle within the framework to be able to choose the best technique to solve the problem that you are interested in.

So, I will see you again in the next lecture on the use of a framework for solving data science problems.

Thank you.

Data Science for Engineers
Prof. Raghunathan Rengasamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 30
Solving Data Analysis Problems – A Guided Thought Process

In the previous lecture, we looked at data analytic problem types, the types of data analytics problems that one typically solves and we also looked at the various techniques that have been used. Though not a comprehensive list gives you a variety of techniques that people are looking at to use to solve data analysis problems and then we also asked questions as to why there are so many techniques and I described one way to think about the techniques and the problems that are being solved.

Now, in many cases as far as this data analysis problems are concerned typically you start with a very not well defined problem I would say. In a typical industrial scenario now-a-days there is a feeling that there is lots of data that is around and everyone seems to suggest that you should be able to use this kind of big data to derive some value to your organization. So, the question then is how do I do it, so typically people start by saying lots of data what is it I can do with this data. You know you might simply say I want improve performance or I want to minimize maintenance problems and so on.

So, you could start talking about a class of problems which could be either performance related or improving the operations, doing things on time and so on. So, typically you start with a loose set of words a vague definition of a problem and the data that you have. Now the question really is to then drive your thought process towards something that is codable, something that you can process the data with to derive value to do any problem that you are solving and so on.

While this is to some extent currently a little bit of unstructured process, good data scientists are able to come up with a solution that makes sense and that is relevant for the problem that needs to be solved. So, what we are going to attempt before we show you the techniques is to give an example of such a thought process so, that if possible you could use this kind of thought process for problems that you need to solve.

I just want to mention that this is not set in stone. We are trying to give structure to your thought process when you solve data analysis

problems. So, to do this we are actually going to take a very simple example and then illustrate how you should think about solving data science problems and at the end of it we will come up with a flow chart that might be useful. So, the problem that we are going to deal with is what is called the data imputation problem.

(Refer Slide Time: 04:05)

Data science for Engineers

Example - Data Imputation

- * Readings from five sensors (X_1, X_2, X_3, X_4, X_5) are made available to you (for 100 different tests, check the file, *GTPvar.csv*). The readings are not arranged according to any order.
- * There are some records, though, where there are a few missing readings that are marked *NA*.
- * Your supervisor has asked you if there are any ideas that can be employed to rationally fill the missing values. Can you develop a data analytic approach to answer this question ?

Framework for data science

2

This is a very standard problem that we see in many engineering applications and not only engineering applications in other fields also. The problem is the following. Remember we talked about the matrix as having values for different attributes at different samples. Now, in many cases I might have samples where I have values for all the attributes and there might be samples there I might not have values for some the attributes and so on. This is a very typical scenario when you look at large data that one deals with in real life.

Now, the question is when I have data that is missing in some samples some attribute values that are missing in some samples, is there some way to fill in that data. So that I make the whole dataset complete and ready for further analysis. So this problem is called the data imputation problem.

So, here we pose an example data imputation problem and then we use this problem to kind of explain what this flowchart or the guided thought processes. What we are going to do is since we have not taught any machine learning techniques as yet we are going to use techniques that we have learnt in linear algebra and statistics to illustrate how we come up with this solution and for a more complicated problem, you would also bring in the other machine learning techniques that you

would learn in this course and then use that in this guided thought process also to solve problems that are of interest.

So, let me read the problem. So, it says reading from five sensors, x one takes five, are made available to you for 100 different tests and in the website we also have this file of data which you can use to go through this process on your own the readings are not arranged according to any order. So, basically what this means is you are not in for any time sequence or any other sequence from this data samples.

Now if all the data were available then you could process this data for other activities. However, in this case there are some records which has data that is missing these are marked as not available n a. Now let us assume your supervisor has asked you if you can come up with any ideas that can be employed to rationally fill the missing values. So, the question is can you develop a data analytic approach to answer this question. So, this is a very typical question that we deal with in real problems and there are many solutions depending on the situation that one looks at. Here we gives one example to illustrate the idea of the framework.

(Refer Slide Time: 07:32)

Data science for Engineers

Example - Data Imputation

- STEP 3: Solution Conceptualization
 - Need complete data set for identifying the function
 - Collect records without missing data
 - Assumption: All variables are independent of each other
 - ⇒ no relation exists between the variables
 - For each variable, fill the missing data with the most likely value
- Step 3a: Verify assumption
 - Assumption not satisfied
- STEP 3: Solution Conceptualization
 - Assumption: Variables are inter-related
 - Step 3a: Assumption cannot be verified a priori

Framework for data science

So, I am going to call the first step in any data analysis problem solving as defining the problem. You define the problem in as broad a way as possible. So that it is very very understandable to everyone and this part of problem definition and the second step which is what I want to call as problem characterization are very important steps and in fact, if these are done properly then the solution, uncovering the solution process, becomes easier.

So, in this case the problem definition is actually fill in missing data records very simple. I have some data which is missing I simply need to fill in these missing data records. Now we drill down a little more and we go on to step 2 which is what type of problem is this. Now in this simple case it will turn out to be one type of problem that I am going to talk about, but in more complicated cases it might be a combination of these basic problem types. So, here at least for this example the problem characterization goes like this. So, we see that given part of the information which is the data that is available fill the missing information.

So, one way to think about this is I need to get some knowledge about the missing information from somewhere and the only place that I can get this information is really from whatever I know about this data from whatever I have with me currently. So, I might say the idea is really to somehow relate the missing information or missing data to the known information or known data. So, that seems like a logical way to solve this problem.

The minute we get to this point right here then we understand that this is a function approximation problem where I am going to write this equation where I have x unknown and if somehow I can relate it to the data, known data or x that is known, then whenever I have something missing I could simply put it into this function and as solve for my variable. So, at the highest level it proceeds in a rather general fashion, but you will see you will have to add more and more ideas into solving these problems as we go along.

Now that we basically said it is a function approximation problem where I am going to relate the unknown data as a function of known data, we need to segregate the data record to completely known samples and samples that are unknown. Because what we are going to do is we are going to relate unknown samples to known samples. So, in some sense we have to get a functional form. So, we try and get complete datasets for identifying these function forms.

So, we might say in a solution conceptualization step, it is important to collect records of data which have no missing data. So, this is a complete set of information, so we could possibly derive something out of this complete set of data some information out of this data and then see whether we could use that information to fill in the missing data. At this point we start making assumptions about the data, remember in the previous lecture I said you could have a function approximation problem you could have many types of functional forms you can use and so on and if you think about statistics there are different types of distributions that you can assume the data follows. You could assume the data is completely deterministic, variables are completely stochastic, variables are some combination and so on.

So, there are many types of assumptions that you can make. So, since there is the first course on data analysis and this the first time we are introducing this framework we are going to keep things very simple. Let us take an assumption which is very commonly made in solving these types of problems, we might simply say these variables are let us say independent of each other; that means, what value a particular variable takes does not really affect the value of other variables.

So, that basically means that no relationship exists between these variables and if we make this assumption then the data filling activity could be for each variable you can fill the missing data with the most likely value. So, there are some fundamental assumptions that you are making when you say that you are going to fill the missing data with most likely value, but from a very simple conceptual layman terms this could just be the average of all the values for that particular variable.

So, if I have let us say let us say these are all known data and let us say this is missing data and I have something like this I have something like this and if I let us say want to just fill in this missing data we start and then get this set of data separately where nothing is missing and then basically say the best way to really fill this is to take an average of these two and put it here. So, somehow we are going to talk about this most likely value. So, we will see how we feel the most likely value. But basically what we are essentially saying is when I want to fill this data all I am going to do is I am going to look at the values only in this column and I am not really going to look at values in the other columns because I have assumed that these variables are not related to each other.

Now, in this case, this assumption can quite easily be verified right at this point from your statistics part of the lecture. You would have seen a quantity called correlation coefficient or quantity that measures the correlation between variables that you calculate. So, you could calculate and find out whether these variables are correlated or not and if they are not correlated then this assumption is fine, but if they are correlated then this assumption that no relation exists between the variables it is not a good one to make. So, I want to emphasize that some of the assumptions that we make can be verified right at this stage based on known statistical ideas and other ideas in terms of linear algebra and so on.

So, if you could verify this assumption and then if there was no relationship, then you say the most likely value then you have to define what the most likely value means. There are many ways of actually defining the most likely value the simplest is what I said you could take the average or you could take the median value you could take a

mode and so on. So, there are many ways of actually defining what the most likely value is.

So, let us assume in this case that this assumption is not satisfied then you go back to the drawing board and then say I have to change my assumption. So, this is where what I explained in the last lecture is important. If this assumption is not satisfied this does not mean there is any problem with the correlation calculation that you have done. That is a good calculation do to do anyway. The only problem is that the assumption that these are not related to each other is not a good assumption to make. So, we go and modify that assumption.

So, we could make the assumption that these variables are interrelated, now at this point just at this point I might not be able to verify this though in this case one could strictly make an argument that that you could verify it here itself, but in more complicated cases there are examples where the assumptions cannot be validated right at this point and you go back to this assumption only after you have used some machine learning technique to solve the problem and then come back and validate any case let us say we cannot validate this assumption a priori.

(Refer Slide Time: 17:28)

Data science for Engineers

Example - Data Imputation

- STEP 4: Method Identification
 - Identify relationships using null space
 - Fill in missing values using the notion of pseudo-inverse
- STEP 5: Actualization
 - Implement in R programming language
- STEP 6 : Assess assumptions
 - Use it in intended application to check performance ? ✓
- Solution realized (OR)
- STEP 3:
- STEP 4:
- STEP 5:
- STEP 6:

So, if we assume that there are relationships between these variables, then there is this question of actually coming up with a method that will answer our question as to what are the relationships among these variables. So, this is what we call as method identification. So, in this case remember if I have a matrix of data this we have discussed several times let us say I have m samples in n variables.

So, we said we have samples in rows and variables in columns, we said if there are a lot more samples than variables which is the case here because we have 100 samples and out of those we have picked out samples that are complete in all respect. So, let us assume there are a certain number of samples which are complete we have only 5 variables. So, the number of records where information is complete is going to be lot more than the 5 variables.

So, basically m is greater than n and then if you want to use things that we have learned before to come back and say I want to solve this problem using things that I have learned, you would quickly realize that we can use the notion of rank and null space to solve this problem and why is it that we can use the notion of rank a null space to solve the problem. We said if you want to identify how many of these variables are actually independent the quantity that you should compute is the rank of the matrix which is what we saw in the linear algebra class.

So, if it turns out that if the rank of this matrix m by n matrix, let us say whatever is the number of records that are complete times 5, because we have 5 variables in this problem, if the rank of this matrix turns out to be 2, then we automatically know that there are 3 relationships. And we also know how to identify these relationships. We can use the notion of null space to identify these relationships.

So, let us assume that we have identified these relationships. That means I basically have A_3 equations in the 5 variables that are there in this problem. Now we are ready to solve the actual problem. For example, if there is a record where there are let us say 3 variables that are missing. So, let us take an example here. So, let us say I have this I do not have this, I do not have this, I have this, I have this, sorry 3 variables are missing, so, I do not have this and I have this and this.

Then if you want to fill this data what you do is basically take these two known values and substitute these two values into the 3 equations. So, those are now known so these 3 equations will go from 5 unknowns to 3 unknowns. So, you have now 3 equations in 3 variables then you can basically solve this problem and fill in this data. If for example, there are 4 variables that are missing in a record then that is one case that we have discussed already in the linear algebra framework. We have let us say 4 variables missing then the 1 variable that we know the value for, we can substitute into these 3 equations and then we will end up with 3 equations in 4 unknowns.

So, this is a case where I have a lot more variables than equations. So, there are infinite number of solutions. So, one possible solution that you could use is to use the pseudo inverse and then find a solution to this problem. Similarly if I have for example, 2 variables that are missing, but 3 are available then when I put these 3 values into the 3

equations I will end up with the system of equations where I have 3 equations in 2 variables.

Now if the equations are all perfect, you could pick any 2 equations from this and then simply solve for the 2 variables. But even if there are minor errors in these equations and the equations are not really perfect in terms of just dropping one equation and solving with other 2 equations, you could still use the notion of pseudo inverse again to solve this problem where I have less variables and more equations.

So, this is a case where we said if all the equations are not consistent you might not be able to find solutions, but we know pseudo inverse is a concept that can be used to solve all types of these cases where you have the same number of equations and variables more equations than variables and less equations and variables and so on. So, this we saw in detail in the linear algebra part of the lecture. So, you notice how a general statement of a problem, fill in data, can be fleshed out in a very systematic way and then you can use concepts that you know, right now since you know from this course at least only concepts from linear algebra and statistics.

We have used only those concepts to illustrate solution to this problem. Now this is a generic approach and once you learn more and more machine learning techniques you will be able to fill in more sophisticated techniques for things like this and then say I am going to use this technique because the assumptions that I have made are consistent with that particular technique and so on.

Now this is conceptually saying pseudo inverse and so on. In an actual data analytics problem solution, you have to actually what we call as actualize this solution which is basically implemented in some programming language of choice. You could do it in math lab, you could do it in scy lab, you could do it in r, you could do it in python and so on. So, in this case for this problem you might write an r code and then once you are done with this then you go back and assess the assumption.

So, maybe you could actually take the completely filled in data with your data imputation and use the data set for whatever the intended application is and then look at whether you are getting a performance that you are happy with. Now if you are not getting a performance that you are happy with then you basically do not blame the null space concept, but you say maybe the problem is that these relationships are not linear or if you assume that there is no error or noise in the data maybe that assumption is not valid.

So, we have to go back and look at those assumptions and then maybe you could say it is still a deterministic problem, but there are non-linear relationships. Then you have to figure out how to get those

non-linear relationships or you could say there is a lot of noise. So, I might use some other idea to fill in the missing data and so on. So, you could say if the noise you could you could attribute a particular distribution for that and depending on the distribution you could use the correct technique and so on.

So, if the solution is realized at this point well and good. If not you go back again to making assumptions. So, the step 3 is where we keep going back where we keep refining our assumptions and what our assumptions can be verified right away; we verify whatever assumptions, we have to wait till the final result to verify, we verify, and then if things work out we are happy otherwise we keep this assumption validation cycle till we solve the problem to our satisfaction.

(Refer Slide Time: 27:02)

Data science for Engineers

Conceptual Framework for Solving Data Analysis Problems

- START: Problem Arrival – Whole lot of words. Diffuse problem statement
- STEP 1: Problem Definition – Convert the loose words in to one problem statement (as precise as possible)
- STEP 2: Problem Characterization
 - Define high-level problems and sub-problems that need to be solved maintaining a high-level granularity
 - Develop a dependence diagram
 - Identify the problems and sub-problems as either function approximation or classification problems

Framework for data science

So, in summary I would say the start of all of this is the first step is a problem arrival whole lot of words very diffuse problem statement. Step one is to convert this into one problem statement or set of problem statements as precise as possible and then to solve that problem you do what I would call as problem characterization. So, you break down this high level problem statement into sub problems and you kind of draw a flow process saying if I solve this sub problem then this result I am going to use in this sub problem and so on.

So, you can think of this like a flowchart that you are drawing with these sub problems and in general if possible you get to a granularity level where you are able to identify the class of problem that the sub problems belong to. In this case of this course we are calling these as function approximation or classification problem so you identify these

problem sorry as either function approximation are classification problems.

(Refer Slide Time: 28:14)

Data science for Engineers

Conceptual Framework for Solving Data Analysis Problems

- STEP 3: Solution Conceptualization – Visualization of the solution process through two conceptual devices
 - List assumptions (3a – Assumptions that can be verified a priori)
 - Flowchart
 - Pictures
- STEP 4: Method Identification – Map the elements of the flowchart and pictures into mathematical modules
 - Identify mathematical constructs/algorithms for the elements in the flowchart/picture
 - Identify lacunae – Data scientist to conceptualize method development
 - Develop the solution method map
- STEP 5: Actualization
 - Realize the solution method map in a software environment of choice
- STEP 6 : Assess assumptions and go through steps 3 to 6 if necessary

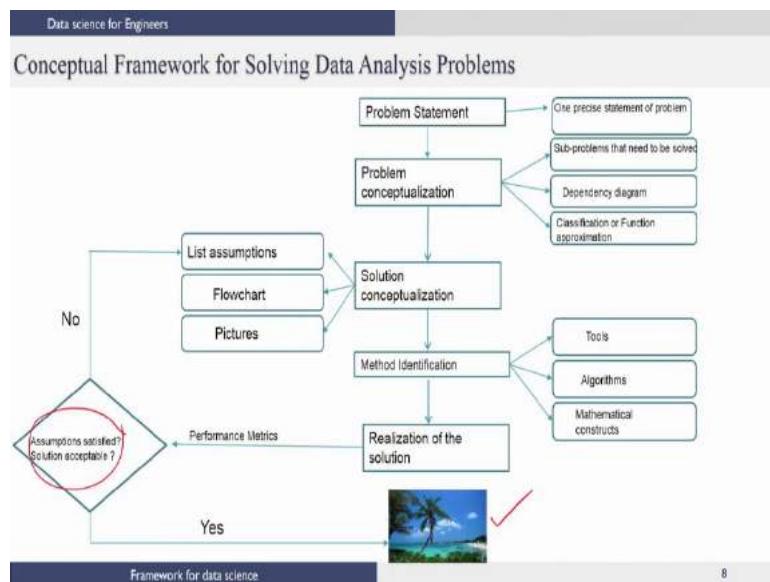
So, this is where you then look at solution conceptualization again you have to make assumptions here. So, you could make assumptions about distributions about linearity and non-linearity the type of non-linearity and so on and here if you if you kind of draw a flowchart and have some pictures in your head then it becomes easier to solve this problem. Then once you conceptualize the solution, then for each of these sub models or sub modules you have to identify a method and the identification of the method should be dictated by the assumptions that you have made.

So, just for classification there are so many techniques which one do you choose. So, we already addressed this in the last lecture you have to look at the assumptions and pick the right method for solution and if it turns out that for the kinds of assumptions that you have made that you do not like any method that is out there then you tweak the existing algorithms to a little bit and then find a method that is useful or that will work for your problem and then once you do this then you basically actualize the solution in some software environment of choice and you basically then get the solution and assess whether the assumptions are good, whether the solution satisfies your requirements and if it does you are done. If it does not you go back and relook at your assumptions and then see how you change or modify your assumptions so that you get a solution that you are comfortable with.

Now, again this step of assessing in the assumption which is basically through the solution that you get is a critical step. Here what we typically do is we partition the data in most cases to data that we

use while we are going through this whole process and then data that has never been used when we were developing the solution and you basically test your algorithms or the flow process that they have come up with on data that has not been seen. So, that is called the test data and this is an important thing to remember and we will emphasize this more when we teach linear regression and classification. So, this is a critical component of assessing assumption. So, this we will see in more detail later.

(Refer Slide Time: 31:04)



So, the whole thing that we have described till now I have as a flowchart here where we start with the problem statement problem conceptualization, solution conceptualization, method identification and realization of solution and finally, when all are assumptions you think are satisfied, the solution is acceptable then you are home clear if not you go back and then redo this till you get a solution that is of value in terms of the problem that you are solving

So, with this the gentle introduction to data science part of the lecture is done. So, we looked at the types of problems, the techniques and why so many techniques and so on and we also provided a framework to guide your thought process and as I mentioned before this is a process that you can use to think about many different problems in a framework in the same way. So that the solution development process becomes easier for you as you go along. You might tweak this framework and then you will have some mental picture or mental framework that you use to solve data science problems which could be this which could be a tweaked version of this or something different, nonetheless the important thing to remember is

that you should think about the problem in a consistent way whenever you solve a problem.

So, what I mean by this is if you use whatever framework it is for a particular type of problem you become aware that you are using such a mental framework when you are solving a problem then you can take the same framework to many different problems then that becomes your thought process for solving data science problems. You do not keep looking at books and say I have to do this, this and this. You have your unique scientific method for thinking about these problems and solving these problems.

So, with this we finished this part of the course and the next set of lectures would be on linear regression which is a type of a technique or a technique for solving function approximation problems and then after that, we will have a series of lectures on some clustering algorithms which can be used largely for classification problems, but can also be used in solving function approximation problems and we will then close the course with a case study and one practical problem description thank you and I hope to see you again when linear regression is started.

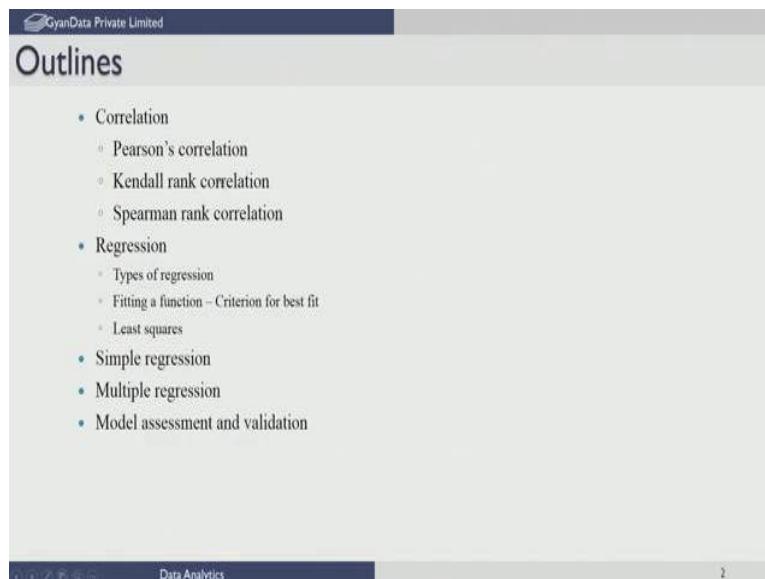
Thank you.

Data Science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture- 31
Module: Predictive Modelling

Welcome to the lectures in Data Analytics. In this series of lectures I am going to introduce to you model building; in particular I will be talking about building linear models using a techniques called regression techniques. So, let us start with some basic concepts. We are going to introduce the notion of correlation.

(Refer Slide Time: 00:38)



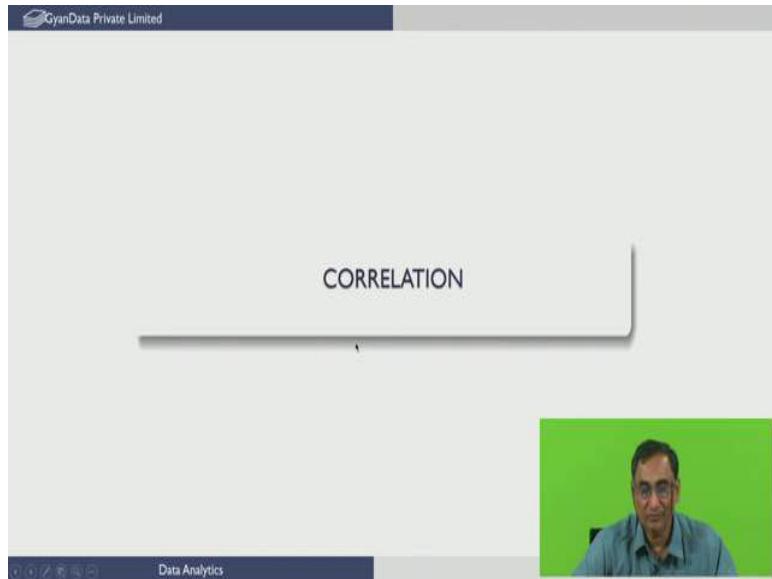
The screenshot shows a presentation slide with a dark blue header containing the GyanData Private Limited logo. The main title is 'Outlines'. Below it is a bulleted list of topics:

- Correlation
 - Pearson's correlation
 - Kendall rank correlation
 - Spearman rank correlation
- Regression
 - Types of regression
 - Fitting a function – Criterion for best fit
 - Least squares
- Simple regression
- Multiple regression
- Model assessment and validation

At the bottom of the slide, there is a navigation bar with icons for back, forward, and search, followed by the text 'Data Analytics' and a page number '1'.

Different types of correlation coefficients that are been defined in the literature what they are useful for this is a preliminary check you can do before you start building models. Then I will talk about regression specifically linear regression and I will introduce the basic notions of regression and then take the case of 2 variables before taking going through multi linear regression where the several input variables and one dependent output variable. Finally, after building the model we would like to assess how well the model performs, how to validate some of the assumptions we have made and so on. So, this is called model assessment and validation.

(Refer Slide Time: 01:16)



So, let us first look at some measures of correlation. We have already seen one in the basic interactive lectures to statistics.

(Refer Slide Time: 01:26)

A screenshot of a presentation slide titled 'Preliminaries'. The slide lists several statistical concepts and formulas:

- n observations for x and y variables (x_i, y_i)
- Sample means \bar{x} and \bar{y}
$$\bar{x} = \frac{\sum x_i}{n} \quad \bar{y} = \frac{\sum y_i}{n}$$
- Sample variances S_{xx} and S_{yy}
$$S_{xx} = \frac{1}{n} \sum (x_i - \bar{x})^2 \quad S_{yy} = \frac{1}{n} \sum (y_i - \bar{y})^2$$
- Sample covariance S_{xy}
$$S_{xy} = \frac{1}{n} \sum (x_i - \bar{x})(y_i - \bar{y})$$

At the bottom of the slide is a navigation bar with icons and the text 'Data Analytics'.

So, let us consider n observations of 2 variables x and y . So, denoted by the samples x_i comma y_i and we can of course, compute the sample means we have seen this in statistics which is just the summation of all values divided by n for x which is denoted by \bar{x} and similarly we can do the sample mean of y which is denoted by \bar{y} . We can also define sample variance, which is nothing but the sum square

deviation of the individual values from the respective means $x_i - \bar{x}$ whole square summed over all the values divided by n or $n - 1$ as the case may be.

So, we define these sample variance S_{xx} and S_{yy} corresponding to the variance of sample variance of x and y. You can also define the cross covariance which is denoted by S_{xy} which is nothing but the deviation of x_i from \bar{x} and the corresponding deviation of y_i from \bar{y} and the product of this we take and sum over all values and divide by n. Notice again that this x_i and y_i order test in the sense that corresponding to the experimental condition ith experiment; we have obtained values for x and y and that is what we have to take you cannot shuffle these values any which way they are known assumed to be corresponding to some experimental conditions that you have set. So, there are n experimental observations you have made.

(Refer Slide Time: 03:03)

Correlation

- Correlation: the strength of association between two variables
- Correlation does not imply causation
- Visual representation of correlation: Scatter grams

Positive trend Negative trend Little or no correlation

Quantitative Metric?

Data Analytics

Now, let us define what we call correlation. Correlation is nothing but the indicates the strength of association between the 2 variables. Of course, if you find the strong correlation it does not mean that a the one variable is a causation, the other is an effect you cannot treat correlation as a causation because there can be a third variable which is basically triggering these two and therefore you can only find the correlation, but cannot assume that one of the variable is a causation and the other is an effect.

We can also before we actually do numerical computation, we can check whether there is an association between variables by using what is called the scatter plot, we have done this before. So, we can plot the

values of x_i on the x axis y_i under y axis and for each of these points and we can see whether these points are oriented in any particular direction. For example, the figure on the left here indicates that the y_i increases as x_i increases there seems to be a trend in this. In particular we can say there is even a linear trend as x_i increases y_i corresponding the increase is in a linear fashion.

This is a positive trend because when x_i increases y_i increases. In the next figure the middle figure we show a case where x_i is as x_i increases y_i seems to decrease and again there seems to be a pattern association between x_i and y_i and this is a negative trend.

Whereas, if you look at the third figure the data that we find seems to be having no bearing on each other. That is x y_i values do not seem to depend in any particular manner on the x_i values. When x_i increases maybe y_i increases for some cases and y_i decreases that is why it is spread in all over the place. So, we can say there is little or no correlation. This is a qualitative way of looking at it, we can quantify this and there are several measures that have been proposed depending on the type of variable and the kind of association you are looking for.

(Refer Slide Time: 05:03)

Pearson's Correlation

- n observations for x and y variables (x_i, y_i)
- Pearson's product-moment correlation coefficient (r_{xy})

$$r_{xy} = \frac{\sum x_i y_i - n \bar{x} \bar{y}}{\sqrt{(\sum x_i^2 - n \bar{x}^2)} \sqrt{(\sum y_i^2 - n \bar{y}^2)}} = \frac{S_{xy}}{\sqrt{S_{xx}} \sqrt{S_{yy}}}$$

- r_{xy} takes a value between -1 (negative correlation) and 1 (positive correlation)
- $r_{xy} = 0$ means no correlation

So, let us look at the most common type of correlation which is called the Pearson's correlation. Here as we started with you have n observations for the variables x and y and we define the Pearson's correlation coefficient denoted by r_{xy} or sometimes denoted by ρ_{xy} by this quantity defined. Where we are essentially taking same thing that we did before the covariance between x and y divided by the standard deviation of x and the standard deviation of y .

The numerator represents the covariance between x and y can also be computed in this manner we can expand that definition we have for the covariance and we can find that it is nothing but the product of $x_i y_i - n$ times the mean of x and the mean of y which represents the covariance of x and y and the denominator represents the standard deviation. We can look at this division by the denominator as what is called normalisation.

So, this value is now bounded. We can show that r_{xy} we take a any value between -1 and $+1$. -1 if it takes a value we say that the 2 variables are negatively correlated if r_{xy} takes a value close to one we say they are positively correlated, on the other hand if r_{xy} happens to value close to 0 it indicates that x and y have no correlation between them. Now, what how we can use this we will see.

(Refer Slide Time: 06:34)

GyanData Private Limited

Pearson's Correlation (Cont.)

- A measure for the degree of linear dependence between x and y
- Cannot be applied to ordinal variables
- Sample size: Moderate (20-30) for good estimate
- Robustness: Outliers can lead to misleading values



Data Analytics

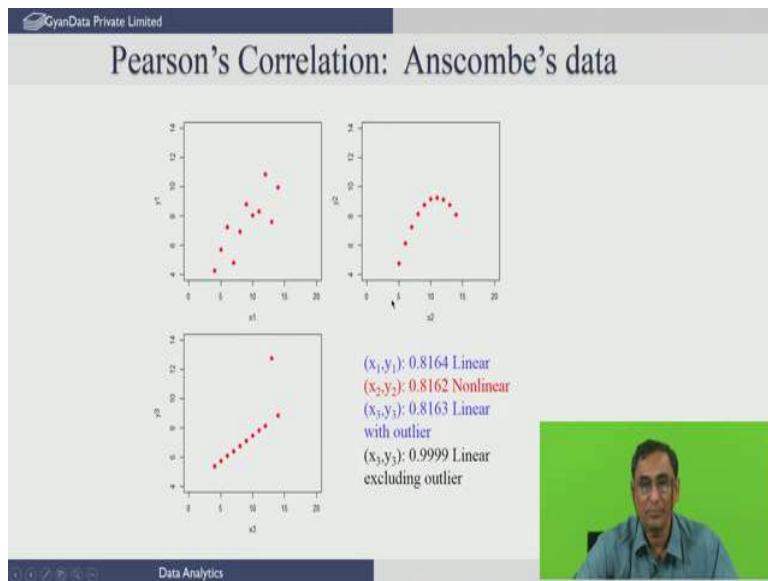
So, this correlation, Pearson's correlation, actually is useful to indicate there is a linear dependency between x and y . For example, if y is a linear function of x then the correlation coefficient or the Pearson's correlation coefficient will turn out to be either close to $+1$ or close to -1 depending on whether y is increasing with x then we say it is a positive correlation as we saw in a figure, if y is decreasing with x linearly then we say it is the correlation coefficient will be close to -1 .

On the other hand if the correlation coefficient is close to 0 all we can conclude from then is perhaps there is no linear relationship between y and x , but perhaps there is non-linear association so, we will come to that a little later. The way it is defined we can also not apply it to ordinal variables which means ranked variable. So, suppose you

have a variable where you have indicated your scale on a scale of say 0 to 10 the let us say the course the those kind of variables are typically you do not apply a Pearson's correlation there are other kinds of correlation coefficients defined for what we call ranked or ordered variable ordinal variables.

Typically in order to get a good estimate of the correlation coefficient between y and x you need at least 20 - 30 points. That is generally recommended and then like the your sample mean or the sample variance standard deviation if there are outliers if there is one bad data point or experimentally you know experimental point which is wrongly recorded for example, that can lead to misleading values of this correlation coefficient. So, it is not robust with respect to outliers just like the sample mean and sample variance we saw earlier.

(Refer Slide Time: 08:24)



So, let us look at some sample examples this is a very famous data set called the Anscombe's data set. Here there are 11 data points for each of this there are 4 data set I have only shown 3 of them, each of them contains exactly 11 data points corresponding to x_i and y_i these points have been carefully selected. In the first one if you look at if you plot the scatter plot you will see that there seems to be linear relationship between y and x in the first data. In the second data if you look at this figure you can conclude that there is a non-linear relationship between x and y and the third one you can say when there is a perfect linear relationship for all the data points except one which seems to be an outlier which is indicated be far away from that line.

So, if I apply Pearson's compute Pearson's correlation coefficient for each of these data sets we find that it is identical, does not matter

whether the, you actually apply into first data set or second data set or the third data set. In fact, the fourth data set has no relationship between x and y and it turns out to be they have the same correlation coefficient. So, what it seems to indicate is that if we apply the Pearson's correlation and we find the high correlation coefficient close to one in this case.

It does not immediately you cannot conclude there is a linear relationship. For example, this is a non-linear relationship and still gives rise to a high value. So, it is not confirmatory in the sense there is a linear you can say it is one way if there is a linear relationship between x and y then the correlation Pearson's correlation coefficient will be high. When I say high it can be - 1 or + 1, but if there is a non-linear relationship between x and y it may be high or it may be low. We will see some data sets to actually show this illustrate point.

(Refer Slide Time: 10:18)

The screenshot shows a presentation slide with the title "Pearson's Correlation (Cont.)". Below the title, there are two bullet points under the heading "Example: Nonlinear".

- Example: Nonlinear
 - $x = 125$ equally spaced values between $[0, 2\pi]$
 - $y = \cos(x)$
 - $r_{xy} = -0.0536$
- Example: Nonlinear
 - $x = 0:0.5:20; y = x^2; r_{xy} = 0.967$
 - $x = -10:0.5:10; y = x^2; r_{xy} = 0.0$

A small video thumbnail in the bottom right corner shows a man speaking. The slide has a navigation bar at the bottom with icons for back, forward, and search, and the text "Data Analytics".

Here are 3 examples. In the first example I have taken 125 equally spaced values between 0 and 2π for x and I have actually computed y as $\cos(x)$. So, this is a relationship between y and x in this case is a sinusoidal relationship. So, if you apply the Pearson's correlation coefficient compute the Pearson correlation coefficient for this data set you get a very low value close to 0 indicating as if there is no association between x and y, but clearly there is a relationship because it is non-linear.

In fact, it is symmetric the points above the 0 line when it is x is between 0 and $\pi/2$ and when it is between $\pi/2$ and π and between π and $3\pi/2$ $3\pi/2$ and 2π they all seem to cancel each other out and

finally, give you a correlation coefficient which is very small, does not indicate imply that there is no relationship between y . All you can conclude from this is perhaps there is no linear relationship between x and y .

Similarly, let us look at another case where this non-linear $y = x^2$, where I have chosen x between 0 and 20 with equally spaced points of 0.5 each 40 points you get and I will compute $y = x^2$ and then compute a Pearson's correlation for this x y data. You find it is a very high correlation coefficient you cannot immediately conclude there is a linear relationship between x and y , you can only say there is a relationship perhaps it is linear maybe it is even non-linear we have to explore further.

If it is close to 0 I would have said there is no linear relationship, but it is in this case it is very high. So, there is some association, but perhaps the association you cannot definitely conclude it is linear it may be non-linear. On the other hand if the data if I chosen my x data between - 10 and 10 symmetrically for between - 10 and 0 $y = x^2$ will have positive values between 0 and 10 $y = x^2$ will have positive values again although x is positive y is positive in this range and between negative values for x y is still positive. So, these will cancel each other out exactly and will turn out that the correlation coefficient in this case is 0 although there is a non-linear relationship between y and x .

So, all we are saying is this if there exists a linear relationship between y and x then the Pearson's correlation coefficient will be either close to 1 or - 1 perfect relationship linear relationship. On the other hand if it is close to 0 you cannot dismiss a relationship between y and x . Similarly if a value is high looking at just the value we cannot conclude that there definitely exists a linear relationship between y and x , you can only say there exists a relationship between y and x . So, let us actually look at other correlation coefficients, you should note that Pearson's correlation coefficient can be applied only to what we call not ranked variables ordinal variables real value variables like we have here.

(Refer Slide Time: 13:18)

GyanData Private Limited

Spearman Rank Correlation

- Degree of association between two variables
- Linear or nonlinear association
- x increases, y increases or decreases monotonically

Data Analytics

10

So, let us look at other correlation coefficients that can be applied even to ordinal variables. Now here is a case where we only look at the again look for degree of association between 2 variables, but this time the relationship may be either linear or non-linear if x increases y increases or decreases monotonically then the Spearman's Rank Correlation will tend to be very high.

So, here is a case when x increases y increases this also is a case when x increases y increases monotonically, but in this case the right hand figure is a non-linear relationship the left hand figure is indicates a linear relationship. Let us apply de ne Spearman's rank correlation apply it to a same data set and see what happens.

(Refer Slide Time: 13:57)

Spearman Rank Correlation

- Spearman rank correlation computation for n observations:

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

d_i is the difference in the ranks given to the two variables values for each item of the data

- Example:

Number	1	2	3	4	5	6	7	8	9	10
X ₁	7	6	4	5	8	7	10	3	9	2
Y ₁	5	4	5	6	10	7	9	2	8	1
Rank X ₁	6.5	5	3	4	8	6.5	10	2	9	1
Rank Y ₁	4.5	3	4.5	6	10	7	9	2	8	1
d^2	4	4	2.25	4	4	0.25	1	0	1	0

$r_s = 0.88$

So, in the Spearman's rank correlation what we do is convert the data even if it is real value data to what we call ranks. So, for example, let us say I have 10 data points in this case x_i is like a somebody has taken a scale between let us say 2 and 10, 1 and 10 and similarly y is also a value between 1 and 10. So, what we have done is looked at all the individual values of x and assigned a rank to it for example, the lowest value in this case x value is 2 and it is given a rank 1 the next highest x value is 3 that is given a rank 2 and so on and so forth.

So, we are ranked all of these points notice that the sixth and the first value both are tied. So, they get the rank 6 and 7 which is the midway, the half of it. So, we have given it a rank of 6.5 because there is a tie. Similarly if there are more than 2 values which are tied we take all these ranks and average them by the number of data points which have equal values and correspondingly you have to in the rank. We also ranked the corresponding y values for example, in this case the tenth value has a rank 1 and so on so forth, eighth value has a rank 2 and so on.

So, we have given a rank in a similar manner now once you have got the rank you compute the difference in the ranks. So, in this case the difference in the rank for the first data point is 2 and we square it, similarly we take the difference in the second data point in the ranks between x_i and y_i which is 2 and square it we get 4.

So, like this we take the difference in the ranks square it and we get the final what we call the d squared values we sum over all values and then we compute this coefficient. It turns out that this coefficient also will lie between -1 and +1 indicating a negative association

and + 1 indicating a positive association between the variables and in this particular case the rank the Spearman rank correlation turns out to be 0.88.

(Refer Slide Time: 16:03)

GyanData Private Limited

Spearman Rank Correlation

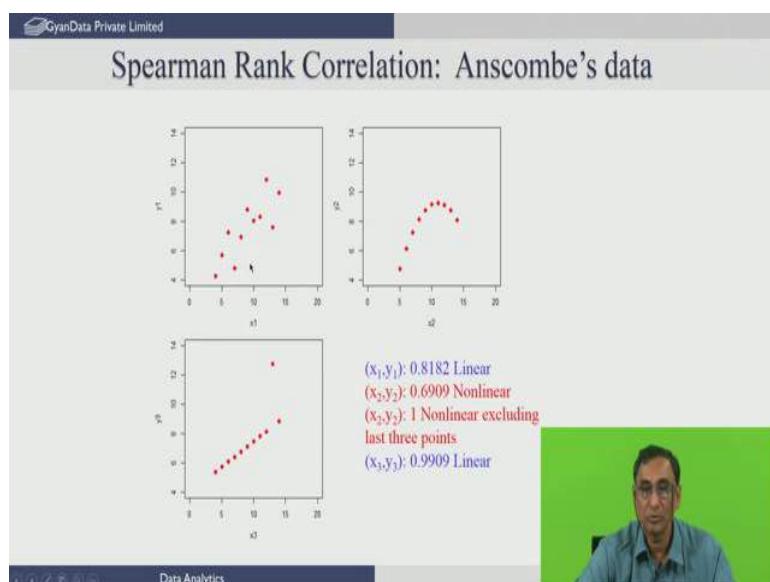
- r_s takes a value between -1 (negative association) and 1 (positive association)
- $r_s = 0$ means no association
- Monotonically increasing $r_s = 1$
- Monotonically decreasing $r_s = -1$
- Can be used when association is nonlinear
- Can be applied for ordinal variables

Data Analytics

12

Let us look at some of the things as I said 0 means no association. When there is the positive association between y and x then the r_s values or the Spearman's thing will be $+1$ like the Pearson's correlation and similarly when y decreases with x then we say that you know the Spearman's rank correlation is likely to be close to -1 and so, on. The difference is between Pearson's and Spearman is not only can it be applied to ordinal variables even if there is a non-linear relationship between y and x the spearman rank correlation can be high it will not likely to be 0, it will have a reasonably high value. So, that can be used to distinguish maybe to look for the kind of relationship between y and x .

(Refer Slide Time: 16:51)



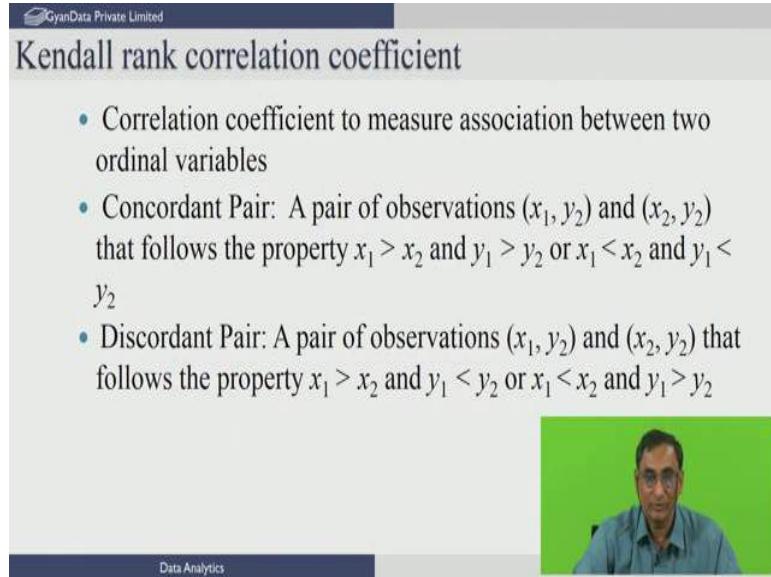
So, let us apply it to the Anscombe data set in this case also we find that the, for the first one the Spearman rank correlation is quite high in the second one also reasonably high. In fact, the Pearson also a sign notice that the Pearson was same for all this and the third one also is fairly high 0.99. So, all of these things it is indicating that there is a really strong association between x and y .

(Refer Slide Time: 17:18)

GyanData Private Limited

Kendall rank correlation coefficient

- Correlation coefficient to measure association between two ordinal variables
- Concordant Pair: A pair of observations (x_1, y_1) and (x_2, y_2) that follows the property $x_1 > x_2$ and $y_1 > y_2$ or $x_1 < x_2$ and $y_1 < y_2$
- Discordant Pair: A pair of observations (x_1, y_1) and (x_2, y_2) that follows the property $x_1 > x_2$ and $y_1 < y_2$ or $x_1 < x_2$ and $y_1 > y_2$



Data Analytics

Suppose we had applied I would suggest that you apply it to the cos x example and y squared x = y square example you will find that the spearman rank correlation for these will be reasonably high it may not be close to one, but it be high indicating there is some kind of a non-linear relationship between them even though Pearson's correlation might be low. So, third type of correlation coefficient that is used for ordinal variables is called the Kendall's rank correlation and this correlation coefficient also measures the association between ordinal variable. In this case what we de ne is a concordant and a discordant pair.

If you look at the values. Compare 2 observations let us say x_1, y_1 and sorry here it should be x_1, y_1 and x_2, y_2 , if x_2 is greater than x_1 and the corresponding y_1 is greater than y_2 then we say it is a concordant pair; that means, if x increases and y also correspondingly increases then these 2 data points are known said to be concordant. Similarly if x decreases if x_1 is less than x_2 i; I am sorry x is increasing. So, x_1 less than x_2 and correspondingly y y_1 is less than y_2 then also we say it is a concordant pair; that means, when x increases y increases or x decreases or y decreases then we say these 2 data pairs are concordant.

On the other hand if there is an opposite kind of relationship, so, if you take 2 data points x_1, y_1 and x_2, y_2 and we say that you know we look at the data points and find that if x_1 is greater than x_2 , but the corresponding y_1 is less than y_2 or if x_1 is less than x_2 , but y_1 is greater than y_2 then we say it is a discordant pair. So, we take every pair of observations in your sample and then assign whether there is a concordant or discordant pair let us take an example and look at it.

(Refer Slide Time: 19:11)

GyanData Private Limited

Kendall rank correlation coefficient

- Kendall rank correlation coefficient

$$\tau = \frac{\text{Number of concordant pairs} - \text{Number of discordant pairs}}{n(n-1)/2}$$

- The pair for which $x_1=x_2$ and $y_1=y_2$ are not classified as concordant or discordant and are ignored.



Data Analytics

So, once we have the number of concordant pairs and number of discordant pairs we take the difference between them divide by n into n - 1 by 2 and that is called the Kendall's τ .

(Refer Slide Time: 19:25)

GyanData Private Limited

Kendall rank correlation coefficient

Example: Two experts ranking on food items

Items	Expert 1	Expert 2						
1	1	1	2	C				
2	2	3	3	C C				
3	3	6	4	C D D				
4	4	2	5	C C C C				
5	5	7	6	C C C D D				
6	6	4	7	C C C C D D				
7	7	5		1 2 3 4 5 6 7				

$$\tau = \frac{15 - 6}{21} = 0.42857$$


Data Analytics

We can take a item here there are about 7 observations let us say 7 different wines or tea or coffee and there are two experts who ranked the taste of the tea or coffee or wine on a scale between 1 to 10. For the first the expert number 1 gives it a rank of 1 and expert 2 also ranks it 1 for the second one the expert 1 ranks it 2 while the expert 2 ranks it in a scale or gives it the value of 3 and so on so forth for the 7 different types of thing.

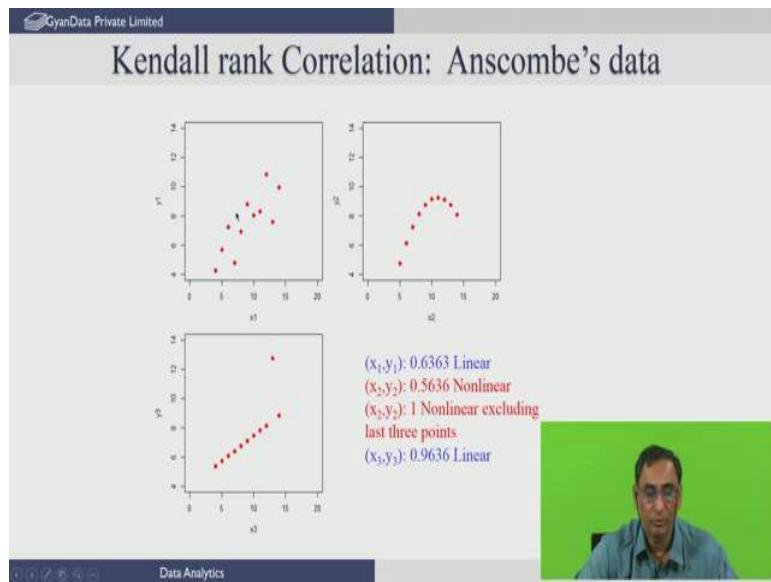
Now, you compare data point 1 and data point 2. In this case experts opinion is that 2 is let us say better than 1, expert 2 also says 2 is better than 1. So, it is a concordant pair. So, 1 and 2 are concordant that is what is indicated here. Similarly if I look at the data point 1 and 3 expert 1 says it is better 3 is better than 1, expert 2 also says 3 is better than 1. So, it is a concordant pair similarly if you look at 2 and 3 both agree in agreement 3 is better than 2, 3 is better than 2 so, it is a concordant.

Let us look at the fourth and the first one looks like expert 1 says it is better, expert 2 also says it is better concordant, but the second and fourth if you compare expert 1 says it is better fourth one is better than the second, but expert 2 disagrees he says the fourth thing is worse than the second one. So, there is a discordant pair of data that is indicated by D. So, 4 and 2 are discordant 4 and 3 are discordant.

Similarly here it says 5 and 1 are concordant 5 2 are concordant and so, on so, forth. So, between every pair n into n - 1 by 2 pairs you will get and we have classified all of these pairs as either concordant or discordant and we find there are 6 discordant pairs and 15 concordant pairs and we can compute the Kendall's τ coefficient. This basically says if this is high then there is broad agreement between the two experts right.

So, basically we are saying y and x are associated with each other also there is a strong association. Otherwise it is not strongly associated or completely if the expert 2 completely disagrees with expert 1 you might get even negative values. So, the high negative value or high positive value indicates that the 2 variables x and y in this case are associated with each other. Again this can be used for ordinal variables because it can work with ranked values here as we have seen in this example.

(Refer Slide Time: 21:56)



So, again if we apply it to Kendall's rank to this Anscombe data set we find that although it has decreased for this linear case the value has decreased as compared to the Pearson and Spearman correlation coefficient, it still has a reasonably high value. High in this case typically you in experimental data you cannot expect to get a value I mean beyond 0.6 or 0.7. You should consider yourself fortunate typically because we rarely know the nature of the relationship between variables we are only trying to model them.

So, in this case non-linear relationship you find again a reasonably high correlation coefficient for Kendall's rank and the last one again it is linear perfectly linear. So, you are getting very high association between them. So, really speaking you can actually use this to get a preliminary idea before you even build the model. Of course, for 2 variables it is easy you can plot it you can compute these correlation coefficient and try to get a preliminary assessment regarding the type of association that is likely to be and then try go ahead and choose the type of models you want to build and this is the first thing that we look before we jump into linear regression.

So, see you next class about how to build the linear regression model.

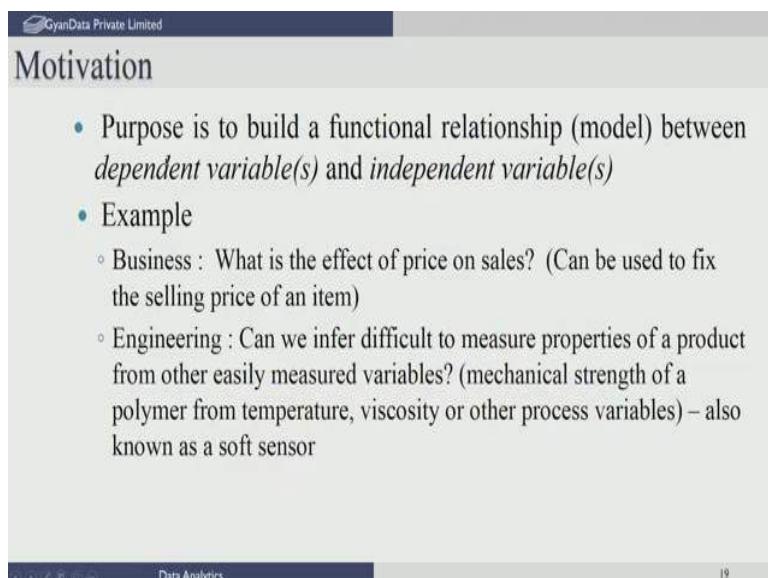
Thank you.

Data Science for Engineers
Prof. Shankar Narasimhan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 32
Linear Regression

Welcome to this lecture on Regression Techniques. Today we are going to introduce to you the method of linear regression, which is very popular technique for analyzing data and building models. We will start with some motivating examples. What is it that regression does?

(Refer Slide Time: 00:37)



GyanData Private Limited

Motivation

- Purpose is to build a functional relationship (model) between *dependent variable(s)* and *independent variable(s)*
- Example
 - Business : What is the effect of price on sales? (Can be used to fix the selling price of an item)
 - Engineering : Can we infer difficult to measure properties of a product from other easily measured variables? (mechanical strength of a polymer from temperature, viscosity or other process variables) – also known as a soft sensor

Navigation icons: back, forward, search, etc.

Data Analytics

19

It is used to build a functional relationship or what we call model, between a dependent variable or variables there may be many of them and an independent variable again there might be more than one independent variable. We will define these variables and how you choose them for the intended purpose a little later, but essentially we are building a relationship between 2 variables you can take it in the simplest case and that relationship we also call it as a model.

So, in literature this is known as building a regression model. We can also call it as identification of a model. Sometimes this goes by the name of identification most popular term is regression.

So, let us take some examples let us take a business case. So, suppose we are interested in finding the effect of price on the sales volume. Why do we want to actually find this effect, we may want to determine what kind of price? We want to set the selling price of an item in order to either boost sales or get a better market share.

So, that is why we are interested in finding what effect does price have on the sales. So, the purpose has to first define what why are we doing this in the first place in this case our ultimate aim is to x the selling price. So, as to increase our market share that is the reason we are trying to find this relationship.

So, similarly let us take an engineering example in this case I am looking at a problem where I am trying to measure or estimate the properties of a product, which cannot be measured directly by means of an instrument easily. However, by measuring other variables, we are trying to kind of infer or estimate this difficult to measure property. A case in point is the mechanical strength of a polymer for it. This is very difficult to measure on line continuously or the other hand process conditions such as temperature, viscosity of the medium can be measured. And from this it is possible to infer provided we have a model that relates the mechanical strength to these variables temperature viscosity and so on. First you develop a model then you can use the model to predict mechanical strength given temperature viscosity.

So, such a model is also known in the literature as a soft sensor or the software sensor and this model is very useful in practice. To continuously infer values of variables which are difficult to measure using an instrument, indirectly you are always inferring it through this model and other variables. So, these are cases where we have the purpose is very clear we are building the model for a given purpose and the purpose is defined depending on the area that you are working in.

(Refer Slide Time: 03:43)

GyanData Private Limited

Regression - Basics

- One of the widely used statistical techniques
- Dependent variables also known as *Response variable, Regressand, Predicted variable, output variable* - denoted as variable/s y
- Independent variable also known as *Predictor variable, Regressor, Exploratory variable, input variable* - denoted as variable/s x

Data Analytics

20

So, regression happens to be one of the most widely used statistical techniques with data and typically there are 2 curbing concepts here. The idea of a dependent variable which is known also in the literature as a response variable or regressand or a predicted variable or simply the output variable. The variable whose output we desire to predict based on the model. So, the symbolic way of denoting this output variable is by the symbol y .

On the other hand we have what is called the independent variable, this is also known in the literature sometimes as the predictor variable or the regressor variable as opposed to predicted and regressand or it is also known as the exploratory variable or very simply as the input variable. We will use the term independent variable for this and dependent variables for the response we will not use the other terms in this talk.

So, the independent variable is denoted by the symbol x typically. So, we have let us for the simple case assume that we have only one variable, which we denote by the variable x the independent variable and we have another variable called the dependent variable, which we wish to predict and we will denote it as y .

(Refer Slide Time: 05:04)

The slide is titled "Regression types" and is part of a presentation by GyanData Private Limited. It contains the following list of classification categories:

- Classification of Regression Analysis
 - Univariate vs Multivariate
 - *Univariate*: One dependent and one independent variable
 - *Multivariate*: Multiple independent and multiple dependent variables
 - Linear vs Nonlinear
 - *Linear*: Relationship is linear between dependent and independent variables
 - *Nonlinear*: Relationship is nonlinear between dependent and independent variables
 - Simple vs Multiple
 - Simple: One dependent and one independent variable (SISO)
 - Multiple: One dependent and many independent variables (MISO)

There are several different classifications and we are just going to give you a brief idea of that. We can have what is called a Univariate regression problem or a multivariate regression problem. The univariate is the simplest regression problem you can come up across, which consists of only one dependent variable and one independent variable. On the other hand if you talk about a multivariate regression problem you have multiple independent variables and multiple dependent variables.

So, to understand the subject it is better to take the simplest problem understand it thoroughly and then you will see the extensions are fairly easy to follow. We can also have what is called linear versus. non-linear regression. Linear regression the relationship that we seek between the dependent and the independent variable is a linear function.

Whereas in a non-linear regression problem the functional relationship between the dependent and independent variable can be arbitrary, can be quadratic, can be sinusoidal or can be any arbitrary non-linear function. And we wish to discover that non-linear function that best describes this relationship that is what forms part of non-linear regression.

We could also classify regression as simple versus multiple simple regression is the case of this single dependent and single depend independent variable also called the SISO system. And multiple regression linear regression is the case when we have one dependent variable and many independent variables or what is called the miso case multiple input single output.

So, these are various ways of denoting the regression problem, we will always look at the simplest problem to start with which is the simple linear regression, which consists of only one independent one dependent variable and analyze it thoroughly.

(Refer Slide Time: 06:52)

GyanData Private Limited

Regression analysis

- Is there a relationship between these variables?
- Is the relationship linear and how strong is the relationship?
- How accurately can we estimate the relationship?
- How good is the model for prediction purposes?

Data Analytics 22

So, the first thing that the various questions that we want to first ask before we start the exercise is do we really think there is a relationship between these variables. And if we believe there is a relationship, then we would not want to find out whether such a relationship is linear or not.

Of course, in linear regression we are going at with the assumption there exists a linear relationship, but you really want to know whether such a relationship linear relationship exists. And how strong is this? How strongly the independent variable affects the response of the dependent variable?

Also we are interested since we are dealing with data that that has random errors or stochastic in nature and we only have a small sample that, we can gather from there from the particular application. We want to ask this question, what is the accuracy of our model? In terms of how accurately we can estimate the relationship or the parameters in this model.

And if we use this model for prediction purposes subsequently how good it is? So, these are some of the questions that we would like to answer, even in the process of developing the regression model.

(Refer Slide Time: 08:06)

The slide is titled "Regression methods" and is presented by GyanData Private Limited. It contains two main sections:

- Linear regression methods**
 - Simple linear regression
 - Multiple linear regression
 - Ridge regression
 - Principal component regression
 - Lasso
 - Partial least squares
- Nonlinear regression methods**
 - Polynomial regression
 - Spline regression
 - Neural networks

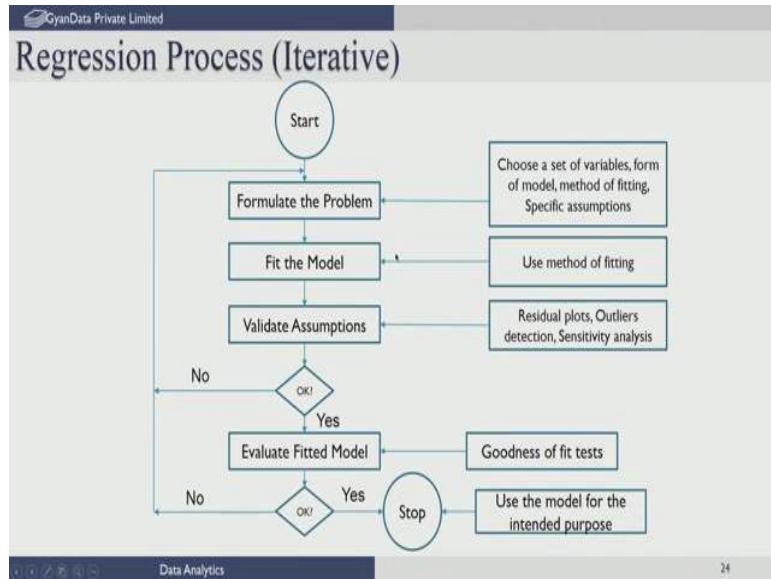
The slide also includes a navigation bar at the bottom with icons for back, forward, and search, and the text "Data Analytics" and "23".

So, there are several methods also, that are available in the literature for performing the regression depending on the kind of assumptions you make and the kind of problems that may you may encounter. As I said the simple linear regression is the very very basic technique, which we will discuss thoroughly. Multiple linear regression is an extension of that for multiple independent variables, but there are other kinds of problems you may encounter, when you have several variables independent variables.

And those have to be attacked differently and there are techniques such as ridge regression or principal component regression, lasso regression, partial least squares and so on so forth, which deals with these kinds of difficulties that you might encounter in multi linear regression.

Of course, in non-linear regression there is again a plethora of methods, I am only listed just a few examples you could have polynomial or spline regression, where the type of equations or functional relationship you specify a priori, you can have neural networks or today a support vector regression. These are methods that are used to develop non linear models between the dependent and independent variable. Now let us take only the simple linear regression and go further.

(Refer Slide Time: 09:15)



So, you have to understand that the regression process it is itself is not a once through process, it is iterative in nature. So, the first question that you should ask us the purpose. Before we even start the regression you ask what is the purpose, what are you trying to develop the model for? Like I said in the business case we are developing the model in order to determine set the price selling price of the thing.

So, you are really interested in how this selling price affects sales. That is the purpose that you have actually got. In the case of a the engineering case we said the purpose is to replace a difficult to measure variable, by other easily measured variables and this model using a combination of the model and other easily measured variables we are predicting a variable, which is difficult to measure online. And then obviously, we can monitor the process using that that parameter.

So, the purpose for each thing has to be well defined, then that leads you to the selection of variables. Which is the output variable that you want to predict and what are the input variables that you think are going to affect the output variable. And so, you choose to set of variables and take measurements, get a sample, do design of experiments, which is not talked about in this whole what we called lecture. So, we will do proper design of experiments in to get what we call meaningful data and once you have the data, we have to decide the type of model. When we say type of model it is a linear model or non-linear model.

So, let us say we have chosen one type of model, then you have to actually choose the type of method that you are going to use in

order to derive the parameters of the model or identify the model as we call it.

Once you have actually done that unfortunately when we use a method it comes with a bunch of assumptions associated you would like to validate or verify, whether the assumptions we have made, in deriving this model are correct or perhaps they are wrong. What this is done by using, what we call residual analysis or residual plots. So, we will examine the residual to kind of judge, whether the assumptions were made in developing the model are acceptable or not.

Sometimes you might have also a few data you may experiment and data may be very bad, but you do not know this a priori, you would like to throw them out they might affect the quality of your model. And therefore, you would like to get rid of these bad data points and only use those good experimental observations for building the model. How to identify such outliers or bad data is also part of the regression. You remove them and then you actually have to redo this exercise.

Finally once you develop the model you want to actually do sensitivity analysis. Is there a if we have a small error in the data how well how much it affects the response variable and so on.

(Refer Slide Time: 15:12)

GyanData Private Limited

Ordinary Least Squares (OLS)

- Fourteen observations obtained on time taken in minutes for service calls and number of units repaired
- Objective is to find relationship between these variables (useful for judging service agent performance)

The scatter plot displays a positive linear relationship between the number of units repaired (X-axis) and the time taken in minutes (Y-axis). The X-axis is labeled 'Units' and has tick marks at 0, 2, 4, 6, 8, and 10. The Y-axis is labeled 'Minutes' and has tick marks at 0, 50, 100, and 120. The data points are approximately as follows:

Units	Minutes
2.5	45
3.5	55
4.5	65
5.5	75
6.5	85
7.5	95
8.5	105
9.5	115
10.0	120
2.0	40
3.0	50
4.0	60
5.0	70
6.0	80
7.0	90

Data Analytics 25

So, this is sensitive analysis you do or if there are many variables we would like to ask this question are all variables equally important or should I discard one of the input variables and so on. So, these are the things that you would do and once you have built the best model that you can from the given data and the set of variables you have chosen then you proceed further.

So, the data that you use in building this model or regression model is also called the training data. You have used the model to train you to use the data to train the model or estimate the parameters of the model. Such that a data set is also denoted as the training data set. Now once you have built the model you would like to see how well does it generalize can it predict the output variable or the dependent variable for other values of the independent variable which you have not seen before.

So, that comes to the testing phase of the model. So, you are evaluating the fitted model using data which is called test data. This test data is different from the training data. So, when you do experimental observations, if you have a lot of data you set apart the sum for training and remaining for testing typically 70 or 80 percent of the data experimental data is used for training or fitting the parameters. And the remaining 20 are used to test the model. This is typically done, if you have a large number of data points.

If you fewer number of observations then you there are other techniques we will actually explain or how to evaluate written models with small samples that you have. So, you first evaluate find out how

well the model predicts on data that it does not seen before and once you have satisfied with it, then you can stop. Otherwise if this model that you have developed even the best model that you have developed under whatever assumptions linear model and so on so forth that you assumed, is not adequate for your purpose you go ahead and change the model type, you may want to actually now consider a non-linear model maybe introduce a quadratic term or you might want to more look at a more general form and redo this entire thing.

It may also turn out that whatever you do you are not getting a good model then maybe you should even look at the set of variables you have chosen and also the type of experiments that we have conducted. So, there could be problems with those that is probably affecting the model development phase. So, when all your attempts have failed you may want to even look at your experimental data that you have gathered. What how did the how did you conduct the experiments, whether there was any problem with that or the variables when you select and did you miss out some important ones.

So, there was quite a lot to regression. What we are going to describe only a small part we are not giving you the entire story, we are only providing a short story on how to formulate or t a model for a linear case. How to validate assumptions and how to evaluate the fitted model. This is basically going to be the focus of the lectures.

So, let us take one small example, which we will use throughout. This is a data of 14 observations small sample, which we have taken on a servicing problem service agents. These service agents, let us say it is like Forbes aquaguard service agent that comes to your house. They go visit several houses and they take a certain amount of time to kind of service the unit or repair it if it is down.

So, they will report the total amount of time taken in minutes let us say for through that they have spent on servicing different customers and the number of units that they have serviced in a given day. So, let us assume that every day the service agent goes out on his rounds and notes the total amount of time he has actually spent and tells at the end of the day reports to his boss the number of units that he has repaired he or she has replied. Let us say that there are several such agents roaming around the city and so on and each of them come back and report.

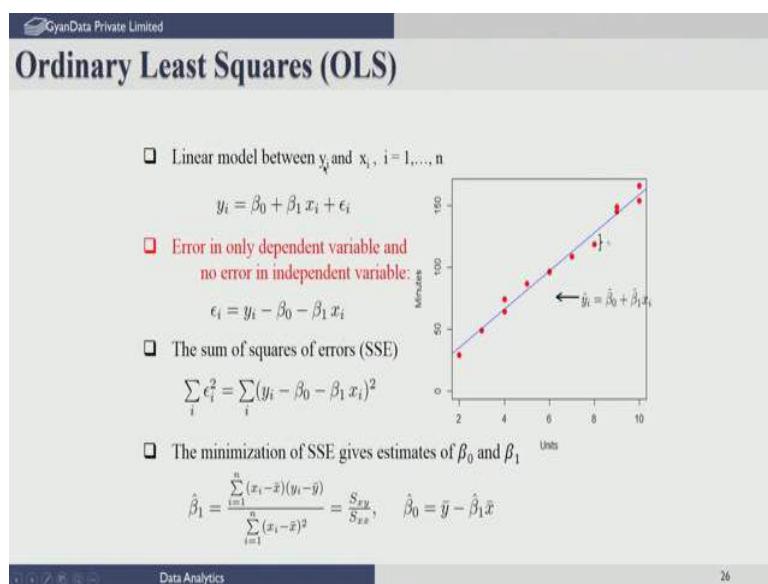
Let us say there are 14 such data points of the same person or multiple persons that you have actually gathered and from this the question that we want to actually answer let us say is given this data suppose as an agent gives you data, you monitor him for week or month on how many how much time they spending, and how many units is repairing every day and want to judge the performance of the

agent service agent. In order to reward or appropriately kind of you know improve his productivity.

So, if you know a relationship between the time taken and number of units repaired which you believe should happen if somebody takes more time and is doing nothing not repairing much, then there is some inefficiency in them maybe he is wasting too much time in between travel or whatever. So, we need to find out right. So, the purpose is to actually judge the service agent performance and do performance incentives in order to improve productivity of these agents. So, we are interested in developing a relationship between number of units and the time taken by something or vice versa now.

For the sake of argument right now I have plotted as we said in A₂ variable case you can visually plot the data scatter plot. So, you plot the data first I have taken units on the x axis and the minutes on the y axis, I will discuss shortly whether we should choose units as the independent variable or minutes as the dependent variable or vice versa, but for the time being I have just plotted it on the x and y axis and look at the spread of data and it looks like there is a linear relationship between the 2 variables. Now we want to build this linear relationship, because from the trend of the data we believe a linear relationship exists let us go ahead with an assumption and just try to build this linear model.

(Refer Slide Time: 18:08)



Now, comes the exact mathematical form in which we state this problem. We have data points x of the independent variable we have n data points in the example n is 14 and y is the dependent variable which we want to use for prediction or whatever purpose that we want for this model.

In this case as I said, given both x and y I would like to rate if some service agent comes and tells this is the time, I have spent and this is a number of units I have repaired and based on it is performance for a month or a week you would like to rate the service agent that is the purpose for building this model.

So, what we do is we have these 14 observations we have taken and we are going to set up the model form. So, as I said we are decided to go with a linear model and the linear model in general can be written like this between y and

x . And this is said that every data point whatever b y b y , whether it is time or units in the previous case, which you have taken as the dependent variable. Can be written in terms of the independent variable as β_0 naught a constant called the intercept term + β_1 times x_i , where β_1 represents what we call the slope.

So, β_1 tells you how a change in x_i affects the response variable y . So, β_0 is a constant it is an o set term, but given x_i the independent variable, it is not possible to get whatever observations we have, observations always contain some error and that error is denoted by ϵ i. So, ϵ i represents an error. Now we have to ask this question what is this error due to. There could be several reasons for this error. This could be because we have not measured x_i precisely or y_i precisely or it could be that the model form we have chosen is perhaps inadequate to explain the relationship, between x and y .

And therefore, whatever we are unable to explain is denoted as ϵ i and it is called a modeling error. So, in this particular ordinary least squares regression that we are going to deal with we will assume whenever we set up the problem x_i the independent variable is assumed to be completely error free. In the sense, we have measured x_i exactly. There is no error in reporting of x_i . On the other hand the dependent variable y_i could contain error. We allow for errors in the reporting of y_i , but the error is not what we call a systematic error it is a random error that is how we are modeling ϵ i or you could also look at ϵ i as a modeling error.

In this particular case where you can say this linear model is only an approximation of the truth and anything that we are not able to explain perhaps can be treated like a random error modeling error. Whatever be the reason the most important thing to note is that this particular model

form re ordinary least squares methodology a formulation does not allow error in the independent variable.

So, when you choose the independent variable one of the things you to do carefully is that you should ensure that this thing is the most accurate among the 2 variables. If you have A₂ variable case you should choose the independent variable as the one which is the most accurate one. In fact, it should be probably error free.

So, let us take the case of the units and minutes. Typically the number of units repaired by a service agent will be reported exactly, because he will have a receipt from each customer and saying that the unit was serviced, you give this back and the total number of receipts that the service agent has gathered precisely represents the number of units service.

So, there cannot be an error unless somebody transcribes this thing, the error in transcription, this can be exactly counted and you would have a precise idea it is an integral number that cannot be an error in this. On the other hand the amount of time taken could vary because of several reasons; one because this guy reported the total time he actually started out on the day and when he returned to the office end of the day. And this could involve not just the service time, but also travel time and depending on the location, the travel time could vary, it could vary from time of day, depending on the traffic, it could also vary because of congestion or a particular even that has happened.

So, the time that has been reported contains other factors that we may not have precisely considered, unless the service agent goes with that stopwatch and measures exactly the time for repair. Typically you will report the total time spent in servicing all of these units including travel time and so on that is the kind of data that you might get.

So, you should regard the minutes as only an approximation, you can not say that is only due to servicing, but also could have other factors which you treat as random disturbance as random error. So, it is better in this case to choose units as the x variable, because that is precise that has no error and y where minutes as the dependent variable.

Notice there might be an argument saying that you know you should always choose the variable which you wish to predict as the dependent variable, need not once you build a model you can always decide to use this model for predicting x given y or y given x.

So, it does not matter how you cast this equation, how you build this model, it is more important that when you apply ordinary least squares, you should actually ensure that the independent variable is an extremely accurate measurement or it represents the truth as closely as

possible. Whereas, y could contain other factors or errors and so on and it is this method is tolerant to it.

So, this goes if on the other hand if you believe both x and y contain significant error, then perhaps you should consider other methods called total least squares or principal component regression that we will talk about later. If positive not in this lecture, but if we have the time we will do it later.

So, essentially what I am saying is that once you have decided based on purpose based on the kind of quality of the measurements, what is the independent dependent variable, then you can go ahead and say given all the observations n observations, what is the best estimate of β_0 and β_1 . As I said that β_0 is the intercept parameter and β_1 is the slope actually geometrically interpret β_0 , β_0 represents the value of y when $x = 0$. So, when you put $x = 0$ and you look at where this line intercepts the y axis this vertical distance is β_0 and the slope, which represents the slope of this regression line that is β_1 so, your estimating the intercept and slope.

So, now what is the methodology for estimating this β_0 and β_1 . So, what we will do is we will do a kind of a thought experiment you give values of β_0 and β_1 and then you can draw this line. So, we will ask different people let us say, values of β_0 and β_1 and draw appropriate lines. Again the line shape that the slope and the intercept will be different depending on what value you propose for β_0 and β_1 . Then once you have done this we will actually go back and find out how much deviation is there between the observed value and the line. In this particular case we will say the observed value let us take this observed value is y_i corresponding to this x_i , which is 8.

Now, the line if this particular equation is correct then this is the predicted value of y , which means for this given value of x_i according to this equation you believe y predicted should be here. And then this deviation between the observed value and the predicted value, which is on this line, the vertical distance is what we call the estimated error.

So, you do not know what the actual error is, but if you propose values for β_0 , β_1 , immediately I am able to derive an estimate for this error which is the vertical distance of the point from that line. We estimate this error for all data points.

So, we compute e_i for every data point y_i using the proposed parameters β_0 and β_1 and the value of the independent variables we have for all the observations. Now what we do is we can say as a metric, what is the best line? We propose that the best line is 1, which minimizes the deviations some square deviations or the distances .

So, overall the data points, we will compute this distance which is geometric distance is nothing, but square of this value we will compute this and sum over all the data points n data points. And we try to find

β_0 and β_1 , which minimizes this sum squared value or minimizes the sum of the vertical distances or the point from that line.

So, the notion of a best t line in the least square sense or the ordinary least square sense is one that minimizes the vertical distance of the points from the proposed line. Now you can, once you set up this formulation, then we can say then who over gives the best β_0 and β_1 will have the minimum vertical distance of the points from that line. And this can be done now analytically instead of asking you now for this β_0 and β_1 I try to solve this optimization problem, which means minimize this, find out β_0 β_1 , which minimizes this and this what is called the unconstrained optimization problem with 2 parameters you differentiate this with respect to β_0 set it = 0 for those called the first order conditions.

Those of you have done a little bit of optimization will know that our calculus, will know all I have to do is differentiate this function with respect to β_0 set it = 0 differentiate this function with respect to β_1 set it = 0 and solve the resulting set of equations. And finally, I will get the solution for β_0 and β_1 , which minimizes this sum squared error. So, the least squares technique uses this as a criterion in order to derive the best values of β_0 and β_1 .

Of course, you can counter by saying I will use some other metric maybe I should have used absolute value. That will make the problem difficult this method was proposed in the late 1700s by Gauss or another person called Legendre and it has become popular as a methodology although in recent years other methods have taken over.

So, the method of least squares is a very popular technique and it gives you parameters analytically for the simplest cases. So, you get β_0 estimated. So, the estimate that you derive is not what you it is not that you should you should treat this estimate as actually the truth it is an estimate from data. Had you given me a different sample maybe I would have got a different estimate remember that. The estimate is always a function of the sample that you are given.

So, we denote such estimates by this hat always implies it is an estimated quantity and the estimated value of β_1 turns out to be the cross covariance between x and y divided by the variance of x. You can prove this. So, remember you this cross covariance is essentially like a Pearson's coefficient. So, the Pearson's coefficient said if the coefficient, Persons correlation coefficient, was close to 1 or - 1, you said that there is you could interpret that there may exist a linear relationship.

Similarly you can see β_1 is a function of that coefficient. It depends on the cross covariance between x and y and β_0 the intercept turns out to be nothing, but the mean of y - the estimated value of β_1 slope parameter multiplied by the mean of x. This is your intercept

parameter. Of course, one could also ask suppose I know that if x is 0 y is 0 I know that a priori.

In this particular case for example, if you do not service any units which means you have not traveled you are not let us say you are on holiday then clearly you would have taken 0 time for servicing. So, I know in this particular case perhaps that that if you process 0 units you should not have taken any time.

So, therefore, the intercept should pass through 0. If you know it and you want you want to force this line to pass through 0 0, the origin, then you should not estimate β_0 you should simply remove this parameter and simply write $y = \beta_1 x$. And in which case the solution for β_1 will turn out to be again S_{xy} by S_{xx} except that this S_{xy} is a cross covariance not about the mean, but about 0. Which means you set $\bar{xy} = 0$ in this expression and you will get σ_{xy} in the numerator over all data points divided by σ_{xi} square not $\sigma_{xi} - \bar{x}$ square.

So, essentially you are taking the variance around 0 and the cross covariance around 0 0 and then you will get the estimated value of β_1 . Of course, β_0 in that case is assumed to be 0. So, the line will pass through 0 0 and you will get another slope. You are forcing the line to pass through 0 0. Remember you have to be careful when you do this, because it will, unless you are sure that should pass through the origin, you should not force this thing you will get a bad t. If you know it and you want demand it it makes physical sense then you are well within your rights to force $\beta_0 = 0$ do not estimate it. That can be done by simply taking the cross covariance and variance around 0 instead of around their respective means .

(Refer Slide Time: 32:25)

 GyanData Private Limited

OLS: Testing Goodness of Fit

- Prediction using the regression equation: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- Coefficient of determination - R^2 is a measure of variability in output variable explained by input variable

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Variability explained by $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
Total variability in y
- R^2 values: Between 0 and 1
 - Values close to 0 indicates poor fit
 - Values close to 1 indicates a good fit (However, should not be used as sole criterion to judge that a linear model is adequate)
- Adjusted \bar{R}^2

$$\bar{R}^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2 / (n-p-1)}{\sum(y_i - \bar{y})^2 / (n-1)}$$

Data Analytics

27

So, this is as far as getting the solution is concerned. So, now, once you get the solution you can ask for every given x_i , what is the corresponding predicted value of y_i using the model. So, you plug in your value of x_i in the estimated model which is using the estimated parameter β_0 and β_1 and I will call this prediction \hat{y}_i it is also an estimated quantity for any given x_i , I can estimate the corresponding y_i using the model.

And geometrically if you actually try to estimate y_i given this point it will fall on this line. You draw the vertical line which intersects this particular regression line and that particular point on that line will represent y_i hat for every point. So, for this point it is actually the corresponding predicted value will lie on this line here if this is the best t line. The blue line represents the best t line in the least square sense.

So, you can do this for any new point which you have not seen before in the test set also. Let us look at some couple of other measures which you can derive from this. We can talk about what is called the coefficient of determination r^2 , which is defined in this manner. It is just $1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$.

So, essentially this particular quantity called r^2 will be between 0 and 1 we can show. So, how to interpret this. The denominator is the variability in y_i what do you mean by that? That is if you are given just y_i and trying to find out how much variance there is there in the data this particular thing divided by n of course, gives you the variance of y .

So, you can say this much variability exists in the data suppose I build the model and try to predict y_i if x_i had a influence on y , then I should be able to reduce it is variability I should be able to do a better prediction and the difference between y_i and \hat{y}_i should be lower. If x_i had a strong influence in determining y .

So, the numerator represents the variability, which is explained by the explanatory variable x or the independent variable x_i . So, if the numerator is approximately equal to the denominator then you basically get 1 and R^2 will be close to 0, the implication of this is x_i has a very little impact on explaining y and probably there is no relationship between y and x . If on the other hand if the numerator is close to 0 and then you get R^2 close to 1 it implies that the x_i can explain the variation in y_i , which means there is a strong relationship between x_i and y_i .

So, values close to 0 indicates a poor t, values close to one indicates a good t, but the problem does not end there. If you get R^2 close to 1 you should not conclude your job is done and the linear

relationship is good and so on. And the Anscombe data for example, when we saw last class, if you try to find the Anscombe data for the 4 datasets you will get all r squared close to one and that does not mean that the linear model is good.

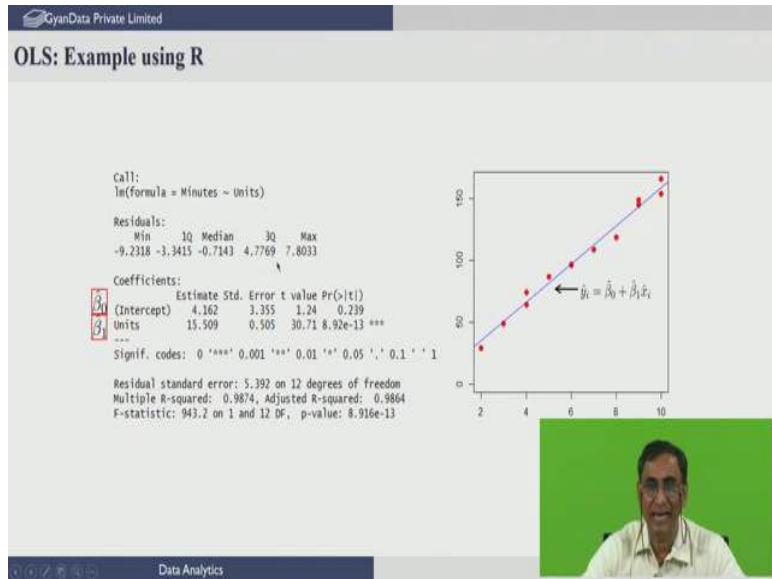
You should look at other measures before you conclude, conclusively determine, that a linear model is good a. A value close to one is a good starting point. Yes you can now be a little bit assurance, you get that the linear model perhaps can explain the relationship between y_i and x_i . There is also something called adjusted r squared, which will come across which is essentially this. If you look at the denominator, you can say that if you try to estimate a constant value. Suppose you say x_i has no influence and dropped that $y\beta_1$ and try to estimate β_0 in the least square sense. You will find that the best value of β_0 is actually best estimate is just \bar{y} .

So, you can regard the denominator as fitting a model with just the parameter β_0 . On the other hand the numerator you have used 2 parameters to fit the model. Whenever you use more parameters typically you should get a better t. So, generally the numerator value is obtained, because you have used 2 parameters. Whereas, the denominator you used only 1 parameter so, you have to account for the fact that, you have used more parameters to obtain a better t and not because there is a linear relationship between y_i and x_i .

So, you should go back and account for this what we call the number of parameters you have used or the number of degrees of freedom that is used in estimating the numerator. For example, you have n data points and in this case the p = 2 parameters. So, n - I am sorry p = 1, which happens to be only β_1 . So, n - 2 would represent the number of degrees of freedom used to estimate this numerator variability whereas, n - 1 is used to estimate the denominator variability, because you have used only the parameter β_0 for denominator whereas, you used 2 parameters to estimate the numerator.

So, you should adjust this by dividing the number of degrees of freedom and the adjusted R square essentially makes this adjustment and give it is different from R squared, but it is a more accurate way of what I call judging whether there is a good linear good model between the dependent and independent variable. And in this case p = 1, because I have only 1 explanatory variable, but this can extend to many independent variable case where p is the number of independent variables you have chosen for fitting the model.

(Refer Slide Time: 38:31)



Finally we will end with the R command. The R command for fitting a linear model is just called lm, if you have loaded the data set and then you say that what kind of what you call variable is the independent variable and what is the dependent variable, you indicate. In this case we have indicated the minutes as the dependent variable and units as the dependent variable and these are variables that forms part of the data set, they are defined as these variables and therefore, you are using them.

So, loading of the data set you would have already seen, lm is the one that you used to build the model, you indicate what is the dependent and independent variable. And then you will get an output that is given here first you will get the range of residuals, which I said is the estimated value of ϵ i for all the data points in this case all the 14 residuals are not given the max value min value the first quartile third quartile in the median are given here.

And I will only now look at 2 parameters the β_0 , which is the first the intercept is called the β_0 estimated values here and the slope parameter the estimated values 15.5 for this particular data set. Now I will also now only focus on this particular line, which talks about the R squared value, which we explained to judge the quality of the model it is a very high R squared you get or the adjusted R squared.

So, from this we can conclude maybe a linear model is explains their relationship between x and y very precisely, but we are not done yet we have to do residual analysis we have to do further what you call plots in order to judge and conclude that linear model is adequate. We will do this and the other things that outputs that are gives as in the

subsequent lectures I will explain them. And, we will see you in the next lecture.

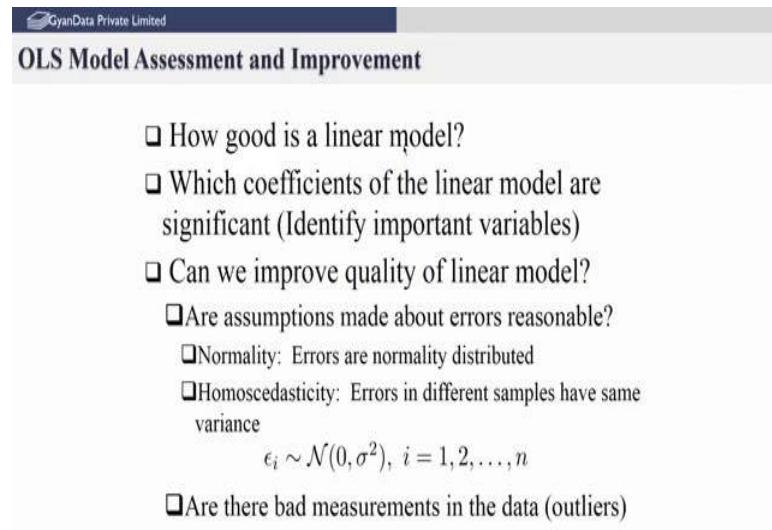
Thank you.

Data Science for Engineers
Prof. Shankar Narasimhan
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 33
Model Assessment

In the previous lecture, we saw how to fit a linear model between two variables x which is the independent variable and y which is the dependent variable using techniques called regression and in this particular lecture we are going to assess whether the model we have actually fitted is reasonably good or not. There are many methods for making this assessment, we will look at some of these. So, what are the useful questions to ask when we fit a model? The first question to ask is whether the linear model that we have fitted is adequate or not, Is good or not. If it is not good then perhaps we may have to go and fit an non-linear model. So, this is the first step that you will actually test whether the model is good or not?

(Refer Slide Time: 01:06)



The screenshot shows a presentation slide with a dark blue header containing the GyanData Private Limited logo. The main title is 'OLS Model Assessment and Improvement'. Below the title is a bulleted list of questions:

- ❑ How good is a linear model?
- ❑ Which coefficients of the linear model are significant (Identify important variables)
- ❑ Can we improve quality of linear model?
 - ❑ Are assumptions made about errors reasonable?
 - ❑ Normality: Errors are normally distributed
 - ❑ Homoscedasticity: Errors in different samples have same variance
$$\epsilon_i \sim \mathcal{N}(0, \sigma^2), i = 1, 2, \dots, n$$
 - ❑ Are there bad measurements in the data (outliers)

Then even if you fit a model you may want to find out which coefficients of the linear model are relevant. For example in the one variable case that we saw one independent variable the only 2 parameters that we are fitting are the intercept term β_0 and the slope term β_1 . So, we want to know whether we should have fitted the intercept or not whether we should have taken it as 0. When we have several independent variables in multilinear regression we will see that

it is also important to find out which variables are significant, whether we should use all the independent variables or whether we should discard some of them.

So, this particular test for finding which coefficients of the linear model are significant is useful not only in the univariate case but more useful in multi linear regression, where we would not identify important variables. Suppose, the linear model that we fit is acceptable then we would not actually see whether we can improve the quality of the linear model. When fitting linear model using the method of least squares we make several options about the errors that corrupt the dependent variable measurements.

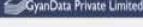
So, are these assumptions really valid? So, what are some of the assumptions that we make about the errors that corrupt the measurements of the dependent variable. We assume that the errors are normally distributed. Only this assumption can actually justify the choice of the method of least squares. We also assume that the errors in different samples have the same variance. So, this is called homoscedasticity assumption. So, we are assuming that the errors in different samples are also having the same variance.

In general, the these two statement assumptions about the errors that they normally distributed with identical variance can be compactly represented by saying that ϵ_i the error corrupting measurement i is normally distributed with zero mean and σ^2 variance. Notice the σ^2 is same and does not depend on i which means it is the same for all samples $i = 1$ to n that is the assumption we are making when we use the standard method of least squares.

Now, we also assume that all the measurements that we have made are reasonably good and there are no bad data points or what we call outliers in the data. We saw that even when we are estimating a sample mean, one that data can result in a very bad estimate of the mean. So, similarly in the method of least squares if we have one bad data point, it can result in a very poor estimate of the coefficients. So, we want to remove such bad data from our data set and improve maybe fit a linear model only using the remaining measurements and that will improve the quality of the linear model.

So, these are some of the things that we need to actually verify. These assumptions what we are made about the errors whether they are reasonable or not if there are bad data, can be remove them or not. And so, we will look at the first two questions in this lecture which is to assess whether the linear model that we are fitted is good and how do we decide whether the coefficients of the linear model are significant.

(Refer Slide Time: 04:33)



OLS: Properties of Estimates

- ❑ Both $\hat{\beta}_0$ and $\hat{\beta}_1$ estimates are unbiased

$$E[\hat{\beta}_0] = \beta_0, \quad E[\hat{\beta}_1] = \beta_1$$

- ❑ Variance of the estimates

$$\text{var}[\hat{\beta}_1] = \frac{\sigma^2}{S_{xx}}, \quad \text{var}[\hat{\beta}_0] = \sigma^2 \frac{\sum x_i^2}{n S_{xx}}$$

- ❑ Estimate of σ^2

$$\hat{\sigma}^2 = \frac{\sum (y_i - \hat{y}_i)^2}{n-2} = \frac{\text{SSE}}{n-2}$$

- ❑ Distribution of slope estimate $\hat{\beta}_1 \sim \mathcal{N}(\beta_1, \frac{\sigma^2}{S_{xx}})$



Data Analytics

31

So, before we start, we need to derive some properties of these estimates that we have derived. Remember, that the coefficients of the linear model that we are fitted which is the intercept term β_0 and the slope term β_1 . These are obtained from data from the sample of data that you are given. We have indicated that these are estimates and not the true values by putting this caret term on top of each of these symbols which means that this is an estimate $\hat{\beta}_0$ is an estimate of the true β_0 and $\hat{\beta}_1$ is an estimate of the true β_1 which we do not know.

However, we can prove based on the assumption we have made regarding the errors that the expected value of $\hat{\beta}_0$ will be $= \beta_0$. What does it mean? If we were to repeat this experiment, collect another sample of n measurement and apply the method of least squares, we will get another estimate of β_0 . Suppose, we do this experiment several times and we will get several estimates of β_0 let me average all of them and average value of that will tend towards the true value. That is what this expression means that if we were to repeat this experiment several times the average of the estimates that we derive will actually, represent, be a very good representation of the truth.

Similarly, we can show that the expected value of $\hat{\beta}_1$ is equal to the true value β_1 . Notice that β_0 and β_1 are unknown values we can we are only saying that the expected value of $\hat{\beta}_1$ will be true value and the expected value of $\hat{\beta}_0$ will be the true value and such statements are also known as if the estimates satisfy such properties we also call these estimates as unbiased, there is no bias in the estimate of β_0 or β_1 .

The second important property that we need to derive about the estimates

is the variability of estimates. Notice we get different-different estimates of β_0 depending on the sample that we have derived and therefore, we want to ask what is the spread of these estimates of β_0 and β_1 if we were to repeat this experiment. We can show again through based on the assumptions we have made that the variance of β_1 will be $= \sigma^2 / S_{xx}$. S_{xx} represents the variance of x or $x - \bar{x}$ the whole squared summed over all the samples. Whereas σ^2 represent the variance of the error that corrupts the dependent variable y . So, σ^2 is the error variance, S_{xx} is the variance of the independent variable. So, this ratio we can show will be equal to the variance of β_1 .

Similarly, we can show that the variance of β_0 is σ^2 which is the variance of the error multiplied by this ratio the numerator is the sum squared values of all the independent variables, while the denominator represents the variance of the independent variable. In this the S_{xx} can be computed from data, σ of x_i^2 can be computed from data, but we may or may not have knowledge about the variance of the error which corrupts the dependent variable that depends on the instrument that was used to measure the dependent variable.

If you have some knowledge of this instrument accuracy we can take the σ^2 from that, but in most cases data analysis cases we may not have been told what is the accuracy of the instrument used to measure the dependent variable. So, σ^2 also have to somehow be estimated from the data. We can show that we can derive a very good estimate of σ^2 by this quantity that is described here which is nothing but the difference between the measured value y_i and estimated value \hat{y}_i which is obtained from the linear equation.

We have fitted a linear model, so, for every x_i we can predict from the linear model what is the estimate of \hat{y}_i for every sample. Then we can take the difference between the measure and the predicted value of the dependent variable sum squared divided by $n - 2$ that is a good estimate of σ^2 which is the error in the dependent variable. Now, why do we divide by $n - 2$ instead of $n - n$ or $n - 1$? Very simple, \hat{y}_i was estimated using the linear model. It had 2 parameters β_0 and β_1 which represents means that 2 of the data points have been used to estimate β_0 and β_1 and therefore, only the remaining $n - 2$ samples are available for estimating this σ^2 .

Suppose, you had only two samples then your numerator would be exactly 0, because you are more than two samples your variability and that variability is caused by the error in the dependent variable. That is one of the reasons that you are dividing by $n - 2$ because two data points have been used to estimate the parameters β_0 and β_1 . Now, this particular numerator term is also called the sum squared errors or SSE for short and so, $\hat{\sigma}^2$ is nothing, but SSE divided by $n - 2$.

So, from the data after you have fitted the model you can compute this value and compute the SSE and obtain an estimate for σ^2 . So, you do not need to be told the information about the accuracy of the instrument used to measure the dependent variable you can get it from the data itself.

So, now finally, not only we have got the first moment properties of $\beta_0 \beta_1$ as well as the second moment properties which is variance of β_1 and variance of β_0 , we can also derive the distribution of the parameters. In particular β_1 can be shown to be normally distributed. Of course, with because the expected value β_1 is β_1 it is normally distributed with β_1 . The true unknown value of β_1 is the mean and the variance given by σ if you substitute σ^2 here you can finally, show that this is nothing, but σ oh, I am sorry. So, this is unknown σ^2 divided by S_{xx} , σ^2 is essentially here we have derived this σ^2 by S_{xx} is the variance of β_1 .

Now, if you do not know σ^2 you can replace this σ^2 is with this σ^2 SSE by $n - 2$. So, once you have derived the distribution of the parameters we can perform hypothesis testing on the parameters to decide whether these are significantly different from 0 and that is what we are going to do. We can also derive what we call confidence intervals for these estimates based on their distribution characteristics that is the mean and the variance.

(Refer Slide Time: 11:48)

 GyanData Private Limited

OLS: Confidence Intervals on regression coefficients

- 95% two-sided confidence intervals (CI) for $\hat{\beta}_0$ and $\hat{\beta}_1$

$$\hat{\beta}_1 \in [\hat{\beta}_1 - 2.18 s_{\hat{\beta}_1}, \hat{\beta}_1 + 2.18 s_{\hat{\beta}_1}], \quad s_{\hat{\beta}_1} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{(n-2)S_{xx}}} \quad t_{0.025, 12}$$

$$\hat{\beta}_0 \in [\hat{\beta}_0 - 2.18 s_{\hat{\beta}_0}, \hat{\beta}_0 + 2.18 s_{\hat{\beta}_0}], \quad s_{\hat{\beta}_0} = s_e \sqrt{\frac{\sum x_i^2}{n S_{xx}}}$$

$$s_e = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{(n-2)}}$$

Now, the first thing we will do is to develop confidence intervals. Confidence intervals simply says what is the interval within which the true value unknown values likely to be with 95 percent confidence or 90 percent confidence you can decide what size confidence interval

size you need to have and correspondingly you can obtain the interval from the distribution.

So, if you want a 95 percent confidence interval also known as CI and it is two sided because it could be either to the left of this estimated value or to the right of the estimated value. So, we are obtaining the 95 percent confidence interval for β_1 from its distribution. You knowing it is normally distributed with some unknown variance. So, that we can actually derived from the from this particular range which is the estimated value of β_1 , which is $\hat{\beta}_1 +$ or - 2.18 times these standard deviation of $b \hat{\beta}_1$ estimated from the data.

Notice this is very similar to the normal thing which says that the true value will between estimate + or - 2 times the standard deviation. The reason why we have 2.18 instead of 2 is because we are no longer obtaining the critical value from the normal distribution, but from the t distribution because σ^2 is estimated from the data and not known up priori.

So, the distribution slightly changes it is not the normal distribution, but the t distribution and that is what we are pointed out here. The steep one 2.18 is nothing, but the critical value 2.5 percent critical value, upper critical value with 12 degrees of freedom. Why 12 degrees of freedom because, you have in this particular example we had fourteen points and we used two of the points for estimating the two parameters. So, $n - 2$ is the degrees of freedom with represents 12. In general, depending on of number of data points this value 2.18 will change. So, that changes is the degrees of freedom of the t distribution from which you should pick the upper and lower critical value. So, lower critical value is - 2.18, the upper critical values 2.18, 2.5 percent. So, the overall is 5 percent. 90 no, this confidence interval represents the 95 percent confidence interval for β_1 .

So, what all we are going to state is that the β_1 to unknown β_1 lies within this interval with ninety five percent confidence that is what we are saying. β_1 can be estimated from data s $\hat{\beta}_1$ can be estimated from data. So, you can construct the confidence interval. Similarly, you can construct the 95 percent confidence interval for β_0 from its variance. So, we are doing the same thing $\hat{\beta}_0 +$ or - 2.18 times standard deviation of $\hat{\beta}_0$ estimated from data which is what we call s $\hat{\beta}_0$.

Remember, s $\hat{\beta}_0$ is σ^2 which is estimated from data multiplied by this square root of $\sum x_i^2$ divided by n times Sxx which is nothing, but the square root of what we have derived in the earlier thing with σ^2 replaced by the estimated quantity. That is all this these two terms represents s $\hat{\beta}_1$ and s $\hat{\beta}_0$. So, having constructed this 95 percent confidence interval you can also use this for testing whether β_0 is unknown $\beta_0 = 0$ or the unknown $\beta_1 = 0$ or not which is what we will do.

(Refer Slide Time: 15:29)

GyanData Private Limited
OLS: Hypotheses test on regression coefficients

- ❑ In order to check if linear model fit is good or not we can test whether estimate $\hat{\beta}_1$ is significant (different from zero) or not
- ❑ Null hypothesis $H_0 : \beta_1 = 0$
- ❑ Alternative hypothesis $H_1 : \beta_1 \neq 0$
- ❑ Null hypothesis implies $\hat{y}_i = \hat{\beta}_0 + \epsilon_i$ ← Reduced Model
- ❑ Alternative hypothesis implies $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \epsilon_i$ ← Full Model
- ❑ Reject null hypothesis if CI for $\hat{\beta}_1$ includes 0
- ❑ Similarly if CI for $\hat{\beta}_0$ includes 0, then intercept term is insignificant

Data Analytics 33

So, let us look at why would want to actually do this hypothesis test. We have fitted linear model assuming that you know that there is a linear dependency between x and y and we have obtained an estimate of $\hat{\beta}_1$. Also we have also fitted an intercept term. We may want to ask should the intercept term significant. Maybe the line should be pass through 0, 0 the origin, maybe the y variable that is not depend on x_1 in a significant manner which means $\hat{\beta}_1$ is approximately equal to 0 that unknown β_1 is exactly equal to 0 although we have got some estimate for β_1 non zero the estimate for $\hat{\beta}_1$.

So, the null hypothesis what we want to test is $\beta_1 = 0$, versus the alternative that $\beta_1 \neq 0$. If $\beta_1 = 0$ it implies that the independent variable x has no effect on the dependent variable, but on the other hand if you reject this null hypothesis we are concluding that the independent variable does have some effect on the dependent variable.

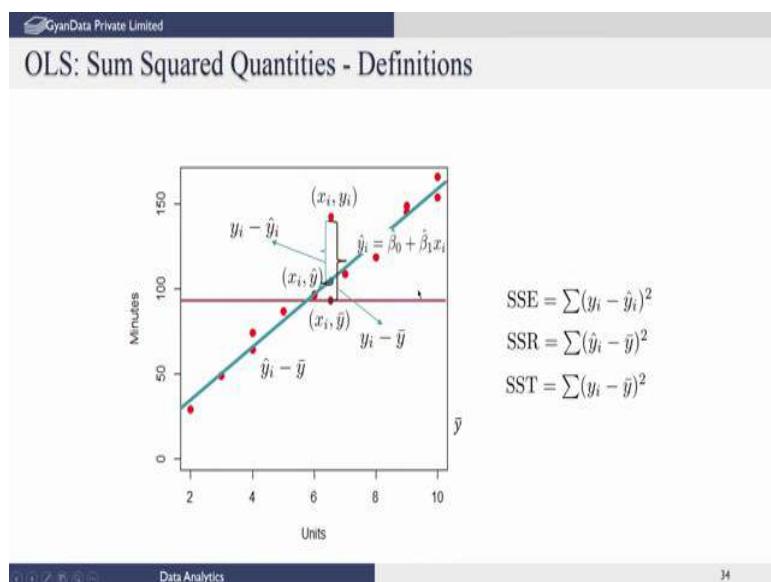
So, this particular hypothesis test can be also reinterpreted as the null hypothesis implies $\beta_1 = 0$ which means what we are doing is only at of $y_i = a$ constant whereas, if we accept in or reject the null hypothesis then we are actually fitting a linear model with β_0 and β_1 present. So, the null hypothesis represents the t of a reduced model which involves only a constant whereas, the rejection of the null hypothesis or the alternative hypothesis implies that we believe there is a linear model that relates y to x.

So, between these two models we want to pick whether the reduced model is acceptable or maybe the full model is to be accepted and the reduced model should be rejected that is what we are doing when we test this hypothesis $\beta_1 = 0$ versus $\beta_1 \neq 0$. Remember, the β_1 can be the positive or negative and that is why we are doing a two sided test.

So, we can do it 2 ways we can actually reject the null hypothesis if the confidence interval for β_1 includes 0. So, notice that we have constructed the confidence interval for β_1 . So, this term $\beta_1 - 2.18$ maybe negative and this maybe positive in which case the interval includes 0 and then we have to definitely we will might make a decision that that β_1 is insignificant and actually true $\beta_1 = 0$. On the other hand if both these quantities if the interval is to the left of 0 which is completely negative or to the right of 0 which means both these quantities are positive, then this interval will not contain 0 and then we can make the conclusion reject the null hypothesis at β_1 equal is 0 which means β_1 is significant. So, from the confidence interval itself is possible to make the reject or the null hypothesis.

So, we can extend this kind of analysis to even test whether β_0 is 0 or not. So, if the confidence interval for β_0 , this particular interval, includes 0, then we will say that the intercept term is insignificant otherwise we will say the intercept term should be is insignificant and should be retained in the model. So, let us actually when we do a final example we will see this. There are other ways of performing this test. And we will continue the we will do that also because that is very useful when we come to multilinear regression. In the univariate regression we have only these two parameters, but multilinear regression there are several parameters you will have one corresponding to each independent variable and therefore, there will be lot more hypothesis test you will look. Therefore, will extend this kind of an argument to test for $\beta_1 = 0$ or $\beta_1 \neq 0$ using what is called a F test which we will go through.

(Refer Slide Time: 19:37)



So, before performing the F test to check whether a reduced model is adequate or we should accept the full model we will use some definitions for sum squared quantities. Notice that let us say that we had set of data in this case we have the example of the number of units that were repaired and the time taken in minutes to repair the units by different sales person and we had fourteen such data points, fourteen such salesman, who have actually reported the data. So, the red points actually represents the data and the best, the linear t, using the method of least squares using all the data points we got something that is indicated by the blue line.

Now, suppose we believe that constant model is good, then we would have actually fitted this particular horizontal line would be the best t representing \bar{y} the best estimate of constant model is the mean of y for all values of x are prediction best prediction for y_i is the mean value of y_i which means x has no relevance β_1 is 0, so, we will estimate the best constant t for y_i is this mean value. So, the red line represents the best t when we ignore β_1 the slope, the blue line represents the best t of the data when we include the slope parameter β_1 .

Now, let us look at certain sum squared deviation the deviation between y_i and \bar{y} which is the redline best t of the constant. This distance is $y_i - \bar{y}$ and sum squared of all these vertical distances from the point to the red horizontal line constant line that is what we call the SS total or sum squared total which also represents the variance of $y_i - \bar{y}$ the whole squared. All that we have not done is divided by n if we have divided by n or $n - 1$, we have got the variance of y, but this represents sum squared errors in y_i when we ignore the slope parameter, that is another way of looking at it.

The distance between y_i and \hat{y}_i . So, now suppose we assume that the slope parameters relevant then we would have fitted this blue line and for every x_i let us take this $x_i y_i$ corresponding to this independent variable, the predicted value of y_i using this linear model would be the intersection point of this vertical line with the blue line which represented by the blue dot which is what we call \hat{y}_i . And therefore, this vertical distance between the measure and the predicted value is the sum squared errors, is called SSE, $(y_i - \hat{y}_i)^2$ and this is the total error if we include the slope parameters in the t.

So, the difference between these two quantities SS total - SS error will be equal to what is also called the sum squared residual which is nothing, but the predicted value - the mean value y bar sum squared over all the data points. Now, we can show that SST will always great be great greater than SSE because SSE was obtained by fitting two parameters there for you should be able to reduce the error ok, maybe marginally, but you will be always able to be able to reduce the errors.

So, SS total is the will always be greater than SSE and therefore, this difference SSR will also be positive all of these a positive quantities. Now, one can you separate SS total as the goodness of t if we assume a constant model, we can interpret SSE as the goodness of t of the linear model and therefore, we can now use this to perform a test. Literally intuitively we can say that if the reduction by including the slope parameter that is SST - SSE is significant, then we conclude it is worthwhile including this extra parameter, otherwise not. This can be converted into hypothesis test formal hypothesis test and that is what is called the F test.

(Refer Slide Time: 24:03)

OLS: F-Test for choosing between models

- F-test for rejecting reduced model
- SST is goodness of fit for reduced model (null hypothesis)
- SSE is goodness of fit for full model (alternative hypothesis)
- F-statistic $F_o = \frac{SST-SSE}{SSE/(n-2)} = \frac{SSR}{SSE/(n-2)}$
- At 5% level of significance reject null hypothesis if $F_o \geq F_{(1,n-2;0.05)}$ (upper critical value of F distribution with 1 and n-2 dfs)
- Note that the numerator has 1 df

So, what we are doing it as I said that SS total is a measure of how good the reduced model is which is reduced model here implies a constant model whereas, the SSE represents how good the linear model if we include this slope parameter. So, we are asking whether the reduced model should be accepted which is the null hypothesis or should be rejected in favour of this alternate which is to include the slope parameter. So, as I said the F-statistic for doing this hypothesis test is to compute the difference in the goodness of fit for the reduced model which is always higher - the goodness of fit SSE for the alternative hypothesis.

So, this represents the sum squared errors for the reduced order model fit, SSE represents the goodness of fit for the alternate hypothesis fit. This difference if it is large enough as I said then we can actually say maybe it is worthwhile going with the alternate hypothesis rather than null hypothesis. So, SSR which is the difference between this should be large enough.

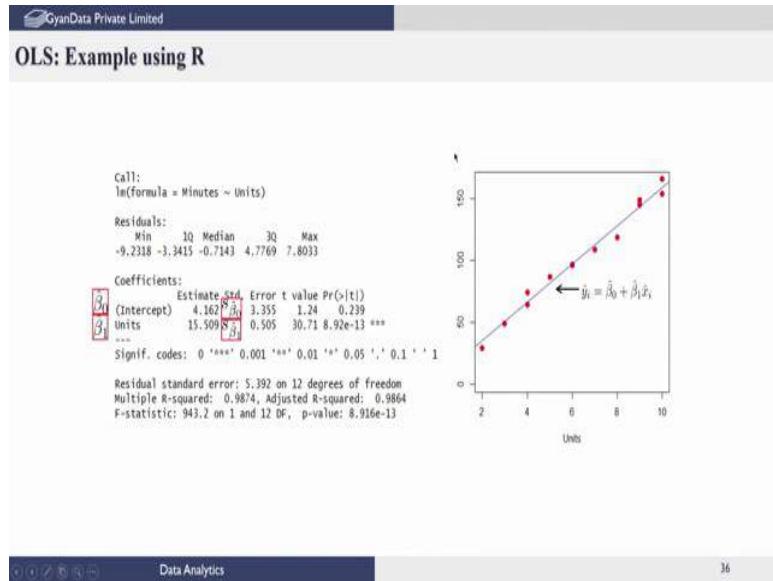
So, normalisation what the denominator represents in some sense of percentage SSE is the error obtained for the alternative hypothesis. Remember because of the different number of parameters used in the model we have to take that into account the numerator SST has $n - 1$ degrees of freedom because we are fitting only one parameter, this has $n - 2$ of freedom because you fitting 2 parameters. So, the difference actually means it is only one extra parameter. So, there is numerator which is SSR has only one degree of freedom which is $n - 1 - n - 2$ whereas, the denominator SSE has $n - 2$ degrees of freedom because it has 2 parameters which is fitted. So, we are dividing the SSE by $n - 2$ the number of degrees of freedom.

So, average sum squared errors per degree of freedom that is what we are saying, and that is your normalisation SSR divided by this normalised is this quantity and we can show formally that is an F-statistic because it is a ratio of two squared quantities and each squared quantities is itself a χ^2 squared variable because it is a square of a normal variable. Therefore, this the ratio of two χ^2 squared and we have seen in hypothesis testing the ratio of two χ^2 squared variable is an F distribution with appropriate degrees of freedom. The numerator degrees of freedom is 1, the denominator degrees of freedom is $n - 2$.

So, if we want to now do a hypothesis test using this statistic F naught we compare F naught with the critical value from the F distribution. Notice F is actually a positive quantity. So, we do a one sided test if we choose the level of significance as 5, 5 percent then we choose the upper critical value from the F distribution with 1 and $n - 2$ degrees of freedom and 5 percent level of significance or what we call the upper critical values probability is 5 percent 0.05. So, once we get this from the F distribution we got this threshold and if the statistic exceeds the threshold then we will reject the null hypothesis and say the full model is better than the reduced model we will accept the full model or we say we reject to reduced model in favour of the full model that the slope parameter is worth including in the model we will get a better t . That is how we actually conclude.

So, there are now several ways for deciding whether the linear model that we have fitted is good or not. We could have used r^2 squared value we said that if it is close to + 1 then we should that is one indicator that the linear model maybe good. It is not sufficient what I call sufficient to conclude, but it is good indicator we can also do the test for β significance of β_1 if we conclude that β_1 is not significant then maybe then a linear model is not good enough we have to find something else or we can do an F test and conclude whether including the slope parameter is significant. So, these are various ways by which we can decide that the linear model is acceptable or not or the t is good. We cannot stop this we have to do further test, but at least these are good in initial indicators that we are on the right track.

(Refer Slide Time: 28:38)



So, let us apply this to the example of repair of or the servicing a problem where we have fourteen data points and the time taken and the number of units repaired by different salesmen are given. So, in this case we have this fourteen points which we have showed we have fitted the data using R. Remember, that `lm` is the function which we should call for fitting a linear model and here we are predicting the dependent variable is minutes and independent variables is units and once we have fitted this using the R function it gives out all of these output and it gives you the coefficient. The intercept term turns out to be 4.16 to the slope parameter turns out to be 15.501.

But, also it also tells you what is this standard deviation, estimated standard deviation, of this parameter which is $S\hat{\beta}_0$ of the intercept it also tells you what is the standard deviation of this estimate for $\hat{\beta}_1$ which turns out to be 0.501 505 all of this calculations from the data using the formulas we have described. Now, once it has given out that we can actually now perhaps construct confidence intervals and find out whether these are significant not or R itself actually tells you something whether these if you run hypothesis test whether you can will conclude whether $\hat{\beta}_0$ is significant or $\hat{\beta}_1$ a significant and that is indicated by what is called this a p value that it is reported.

So, if you get a very high value, t values represents the statistic which have again described earlier. So, it has computed the statistic for you for $\hat{\beta}_0$ and the statistic for testing whether $\hat{\beta}_1 = 0$ or not and it has computed this statistic value and it has compared with the critical value while the distribution, t distribution with appropriate degrees of freedom and concluded that the upper critical or the probabilities 0.239

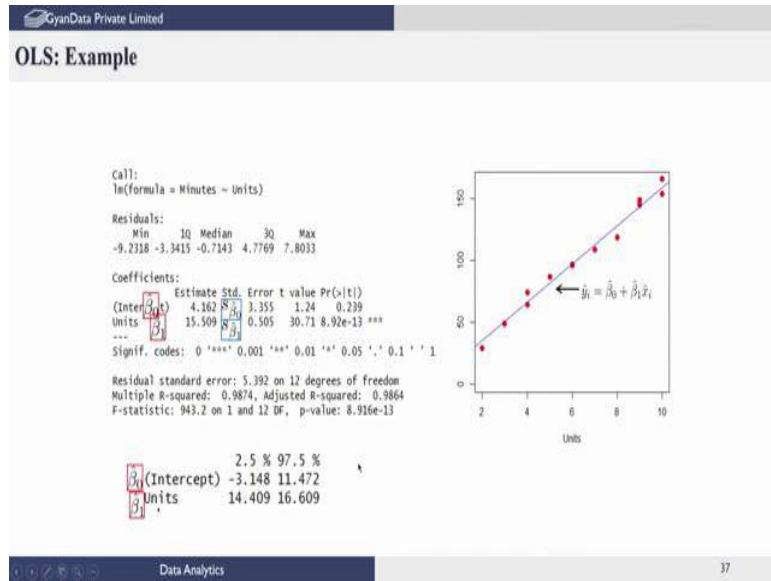
which means if you get very high value for this anything greater than 0.01 or 0.05, it means you should reject the null hypothesis on the other hand if you get a very low value it means you should not reject the null hypothesis, with greater confidence you can reject the null hypothesis.

So, in this case all it is saying is, if you choose a level of significance 0.001 you would not reject the null hypothesis. If you choose 0.05 you will not reject the null hypothesis, if you choose 0.01 as your level of significance you will not reject the null hypothesis. So, that is what the star indicates. At what level of significance will you reject it. Whereas, in the case of β_1 you will reject the null hypothesis which means you will conclude that β_1 is significant even if you choose very low significance value 0.05, 0.01, 0.001 or even lower value. In fact, up to power - 13 you will end up rejecting the null hypothesis. Very low type one error probability if you choose also you will reject the null hypothesis.

So, therefore, you can conclude from these values that β_0 is insignificant which means $\beta_0 = 0$ is a reasonable hypothesis, β_1 is not = 0 is a reasonable hypothesis. Let us go and see whether this makes sense of this data. We know that if the no units are repaired then clearly no time should be taken by this sales repair person. Which means because you have not taken any time for servicing, yes because you have not repaired any units. So, this line technically should pass through 0, 0 and that is what you have said. But, however, we went ahead really and fitted a intercept term, but the tests for hypothesis says you can safely assume β_0 the intercept is 0 it makes physical sense also and we could have only fitted β_1 that is good enough for the data.

So, perhaps you should redo this linear t with β_0 and only using β_1 and you will get a slight different solution and you can test again. So, another way of deciding whether the significant whether the slope parameters significant or not this to look at the F statistic. Notice F statistic is very high and this p values very low which means you will reject the null hypothesis that the reduced model is adequate implying that you should use β_1 including β_1 is very good you will get a better t using β_1 in your modeling. So, the high value of the statistic indicates that it will reject the null hypothesis or a low value of p value for this F this F statistic indicates that you will reject the null hypothesis even at very low significance level.

(Refer Slide Time: 33:40)



You can also construct the confidence interval for β_0 and β_1 and from the earlier thing you say approximately it is estimated + or - 2.18 times the standard error and that is what it is deemed 4.1 + or - 2.18 times 3.35 and that turns out to give the that gives the interval confidence interval - 3.148 to 11.472. That means, with 95 percent confidence we can claim that the true β_0 lies in this interval. Similarly, we can construct the interval confidence interval for β_1 hat, 90 percent confidence interval and it turns out it is 15 + or - approximately two times 0.5 which is 14 and 16.6.

Now, clearly the interval confidence interval for β_0 includes zero and therefore, we should not reject the null hypothesis $\beta_0 = 0$. We should simply accept that β_0 perhaps = 0 whereas, interval for confidence interval β_1 does not include 0. So, we can reject the null hypothesis that β_1 includes 0 and the slope is an important parameter to retain in the model.

Now, all this we have done only for single thing. We will be extending it to the multi-linear case and we will also look at other assumptions, the influence of bad data and so on in the following lecture. So, see you in the next lecture.

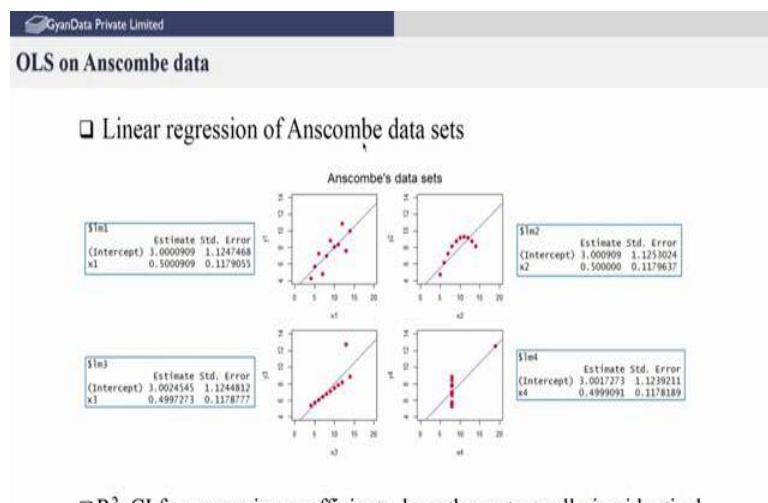
Data Science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 34
Diagnostics to Improve Linear Model Fit

Good morning. In the previous lecture we saw different measures that we can use to assess whether the linear model that we have fitted is good or not. For example, we can use the r squared value, and if we find that the r squared value is close to + 1, we can maybe accept the fact that the linear model is good. We can also check based on the f statistic whether the reduced model is better than the model with the slope parameter. So, if we reject the null hypothesis, there again we may conclude that the linear model is acceptable.

We can also do this by testing the significance of the slope parameter. We can look at the confidence interval for the slope parameter and if that does not include 0, then maybe we can accept a linear model. But all of these measures are not sufficient they only provide a initial indicator. I will show you some data set which shows that that these measures are not completely sufficient to accept a linear model. We will use other diagnostic measure to conclusively accept or reject a linear model fit. So, let us look at a data set provided by Anscombe.

(Refer Slide Time: 01:38)



❑ R², CI for regression coefficients, hypotheses tests all give identical results for all four data sets!

We have seen this before in the when we analyzed statistical measures in one of the lectures, the Anscombe data set consists of 4 data sets. Each one of them having 11 data points, x versus y they are synthetically constructed to illustrate the point. For example, these 4 data sets are plotted x versus y, the scatter plot is given, first data set here second third and fourth. And in all of these if you actually look at it look at the scatter plot we may say that look a linear model is adequate for the first data set, and perhaps for the third data set, but the second data set indicates that the linear model may not be a good choice. A non-linear model or a quadratic model may be a better fit. The last data set is a very poorly designed data set. You can see that the experiment is conducted only at 2 distinct values of x, you have one value of x here for which you have 10 experiments conducted you have got 10 different y values for the same x. And then you have one more experimental observation at a different value of x.

So, you should in this case you should not attempt to fit a linear model with the data. Instead, you should ask the experimenter to go and collect data different values of x, then come back and try to check whether that is valid. Unfortunately, when we actually apply linear regression to these data sets, and then find the slope and the intercept parameter we find that in all 4 cases we get the same intercept value of 3, you can see that all 4 data sets you get a value of 3, and you also get the same slope parameter which is point 5 in all 4 cases.

So, the regression model if you are fit to any of these data 4 data sets you will get the same estimate of the intercept and slope. Furthermore, you get the same standard error of fit which is 1.12 for intercept and point one for the slope. And if you run a confidence interval for the slope parameter, you may end up accepting that this slope is acceptable for all 4 cases. And you may conclude incorrectly conclude that the linear model is adequate. You can actually run the r squared value it will be the same for all 4 data sets. You can run the a hypothesis test whether a reduced model is acceptable compared to a model with the slope parameter, again you will reject the null hypothesis using the f statistic. And you may conclude for all 4 cases, you will get the same identical result that a linear model is a good fit. Clearly it is not so.

One can of course, do scatter plots and try to judge it in this particular case. Because it is a univariate example, but when you have many independent variables, then you have to examine several such scatter plots and that may not be very easy.

So, if you assume there are 100 independent variables, you have to examine 100 such plots of y versus x. And it may not be possible for you to make a visual conclusion from that. So, we will use other kinds

of plots called residual plots, which will enable us to do this whether it is a univariate regression problem or a multivariate regression problem, we will see what these are.

(Refer Slide Time: 04:57)

GyanData Private Limited

OLS: Residual Analysis

□ Questions:

- Do the underlying data satisfy the assumptions on errors (normality, same variance)?
- Is data free of outliers?
- Do some observations exert more influence than others?
- Can the regression equation be improved by using a nonlinear model?

Anscombe's data sets

Data Analytics

40

So, the main questions that we are trying to ask now whether a linear model is adequate. We have some measures we have seen, but they are not adequate, we will use additional things, and when we did the linear regression, we did make additional assumptions although they were not been stated explicitly, we assume that the errors that corrupt the dependent variables are normally distributed and they have identical variance. Only under this these assumptions can you use a least squares method to perform linear regression. That you can at least prove that the least squares method has some nice properties.

So, we do not know whether this is true and we have to verify whether the errors are normally distributed and have equal variance. We also may have a problem of data containing outliers which we may have to remove, and that also we have to solve. Additional questions may be that some observations may have unduly high influence than others and we want to identify such points and perhaps remove them or at least be aware of this. And lastly of course, a linear model maybe inner equates. So, we have to try and t a non-linear model. So, I am going to only address the first 2 questions, whether the errors are normally distributed, whether they have equal variance and whether there are outliers in the data. These 2 things we will address using residual plots. So, let us do this illustrate with the anscombe data set and also other data set.

(Refer Slide Time: 06:27)

OLS: Residual plots

- A straightforward method for assessment of a model is by analysing residuals using *Residual plots*
- Residual definition for OLS

$$e_i = y_i - \hat{y}_i, \quad i = 1, 2, \dots, n$$

➤ Variance of e_i is not same for all data points and also correlated

$$\text{Var}(e_i) = \sigma^2(1 - p_{ii}), \quad p_{ii} = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum(x_i - \bar{x})^2}$$

$$\text{Cov}(e_i e_j) = -\sigma^2(p_{ij}), \quad p_{ij} = \frac{1}{n} + \frac{(x_i - \bar{x})(x_j - \bar{x})}{\sum(x_i - \bar{x})^2}$$

Data Analytics

41

So, one way of assessing, whether they are outliers or whether linear model is adequate or not is using what we call residual plots. And let us see what these residuals are. By definition a residual is the difference between the measured value of the dependent variable - the predicted value of the dependent variable for each sample. So, y_i represents the measured value, \hat{y}_i is the predicted value using the linear regression model that we are fitted. So, that difference is designated as e_i , and it is called the residual. And that is nothing but the vertical distance between the fitted line and the observation point. Now we can try to compute the statistical properties of these residuals, and we will be able to show that the variance of these residuals are not all identical, even though we started with the assumption that all errors corrupting the measured values have the same variance.

But the residual which is a result of the fit will not have the same variance, for all data points. In fact, you can show that the variance of the i th sample is σ^2 which represents the error in the measured value of the dependent variable multiplied by $1 - p_{ii}$; where p_{ii} is defined by this. Notice that p_{ii} depends on the i th sample, numerator depends on the i th sample, therefore p_{ii} depends on the i th sample and varies with sample to sample.

So, the variance of the residual will not be identical for all examples, it is given by this quantity. We also can show that the residuals are not independent even though we assume that the errors corrupting the measurements are all independent. The residuals in the samples are not independent and they have a correlation covariance and that covariance can be shown to be given by this quantity. The reason for the variance not being identical of the residuals or them

being correlated is because you notice that this \hat{y}_i they have actually have here is a result of the regression it depends on all variables all measurements. It is not dependent only on the i th measurement. This predicted value is a function of all the observations, and that because of that it introduces a correlations between the different residuals. And also, imparts different variants to different residuals. And so, having derived this, notice that even if we do not have a priori knowledge of σ^2 square which is the variance of error in the measurements we can estimate this quantity.

We have already seen this in the previous lecture, we can replace this by s^2 by $n - 2$, which is an estimate of this σ^2 and substitute this to get an estimated variance of each residual.

(Refer Slide Time: 09:34)

OLS: Residual plots

- Standardized residual

$$z_i = \frac{e_i}{s_e \sqrt{(1 - p_{ii})}}$$

- If residual variance is estimated from data then
standardized residual has a t distribution with $n-2$ df

We will standardize these residuals, where what we mean by standardization is to divide the residual by its standard deviation estimated standard deviation. All of this can be computed from the data, and therefore, you get for each sample a standardized residual after performing the linear fit which is given by this quantity. Now you can also show that this particular quantity the standardized residual will have a t distribution with $n - 2$ degrees of freedom. Now these statistical properties allow you to now perform test on the residuals, which will what we will use, to identify outliers and also test whether there is set the variances in the different measurements are identical or not.

(Refer Slide Time: 10:21)

The screenshot shows a presentation slide with the following content:

OLS: Residual plots

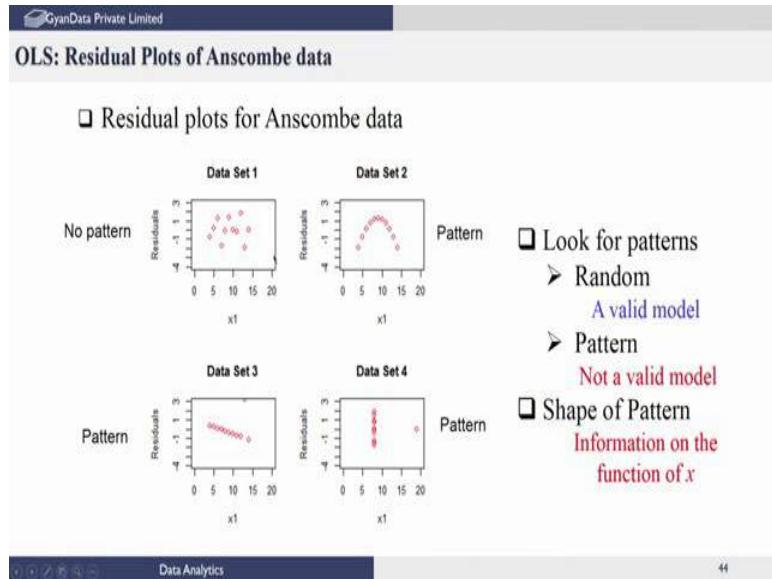
- Residual plot
- Plot of residuals vs predicted (fitted) value of dependent variable
- Residual plots are used for assessing
 - Validity of the linear model
 - Normality of the errors
 - Homoscedastic vs heteroscedastic error

At the bottom right of the slide, there is a video thumbnail showing a man with glasses and a blue shirt, sitting in front of a green screen. Below the video thumbnail is the text "Data Analytics".

So, we will plot the residual what we call a residual plots will we plot the residuals with respect to the predicted or fitted value of the dependent variable remember there is. Only one dependent variable even if there are multiple independent variables we have only one dependent variable, we can plot the residuals with respect to the predicted value of the dependent variable. And the predicted values you obtain after the regression remember.

So, this is called the residual plot what is called the residual versus the fitted or predicted value and this plot is very useful in testing the validity of the linear model, in determining whether the errors are normally distributed, assumptions on errors are ok and whether the variances of all errors are identical or not which is called the homoscedastic case, which means the errors in all measured values are identical or the variance of the error in different measured values are non-identical which is called the heteroscedastic case or heteroscedastic error. So, let us see how each of these how the plot looks for each of these cases.

(Refer Slide Time: 11:28)



Now, let plot the residual plot for the 4 data sets provided by Anscombe. Notice that we have done the regression model regression model we computed all these parameters r squared confidence interval they all turned out to be identical they gave us no clues, whether the linear model is good for all 4 data sets or not. Basically they say they would say that the linear model is adequate.

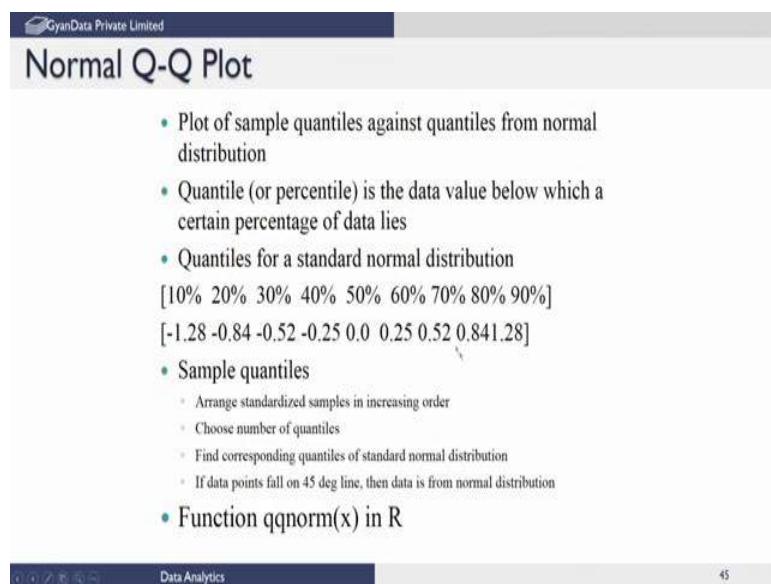
But when we do the residual plot here we have plotted the residual versus the I have plotted with respect to the independent variable, but because it is a univariate case, we have plotted with respect to the independent variable. But technically you should plot the residual with respect to the predicted value of the dependent variable, remember because we presume that the predicted variable is linearly dependent on x, in this case it may not matter the pattern will look the same you can try it out for yourself if you plot the residual with respect to the predicted value of the variable dependent variable. Then you will get this kind of pattern of the residuals for the 4 data sets. The first data set if you look at it exhibits no pattern. The residuals seems to be randomly distributed between this case between - 3 and + 3 and whereas for the second data set there is a distinct pattern. The residuals look like a quadratic like a parabola. And so, therefore, there exists a pattern in the data set 2. For the third data set basically, you can say that there is no pattern, except that are constant more or less linear are constant, there seems to be a small bias because of the slope left in the residuals. Data set 4 as we saw before is a poorly designed experimental data set. All the y values are obtained at a single x value and that is what the residuals are also showing. The 10 of the data points obtained at the same x value are showing different residuals and the one single residual at a different x value showing something.

So, from this you cannot judge anything. All you can say is that the experimental data set is very poorly designed, and you need to get back to the experimenter and ask him to provide a different data set. Now based on this we can safely conclude that data set one clearly a linear model is adequate. All the measures, previous measures also, were satisfied. And now the residual plot also shows a random pattern which means a random or what we call no pattern, then linear model is adequate. Whereas, for data set 2 by look at the residual plot we can conclude that a linear model is inadequate should not be used for this data set.

For the third data set however, we know there is one data point that is lying far away, and perhaps that is the one that is causing all of this slightly a linear pattern here. And if we remove this outlier and retry it maybe this resolve this problem will get resolved, and linear model may be adequate for data set 3. For data set 4 again there is a distinct constant pattern and therefore, we can conclude that linear model should not be used.

In fact, no model should be used between x and y, because y does not seem to be dependent on x here. So, the residual plot clearly gives the game away, and it should be used along with other measures in order to finally, conclude that the linear model that were fitted for the data is acceptable or not. So, in this case data set one certainly will accept data set 2 3 we will have to do further analysis. But for 2 and 4 we will completely reject the linear model.

(Refer Slide Time: 14:59)



The slide content is titled "Normal Q-Q Plot". It includes a bulleted list of points:

- Plot of sample quantiles against quantiles from normal distribution
- Quantile (or percentile) is the data value below which a certain percentage of data lies
 - [10% 20% 30% 40% 50% 60% 70% 80% 90%]
 - [-1.28 -0.84 -0.52 -0.25 0.0 0.25 0.52 0.84 1.28]
- Sample quantiles
 - Arrange standardized samples in increasing order
 - Choose number of quantiles
 - Find corresponding quantiles of standard normal distribution
 - If data points fall on 45 deg line, then data is from normal distribution
- Function `qnorm(x)` in R

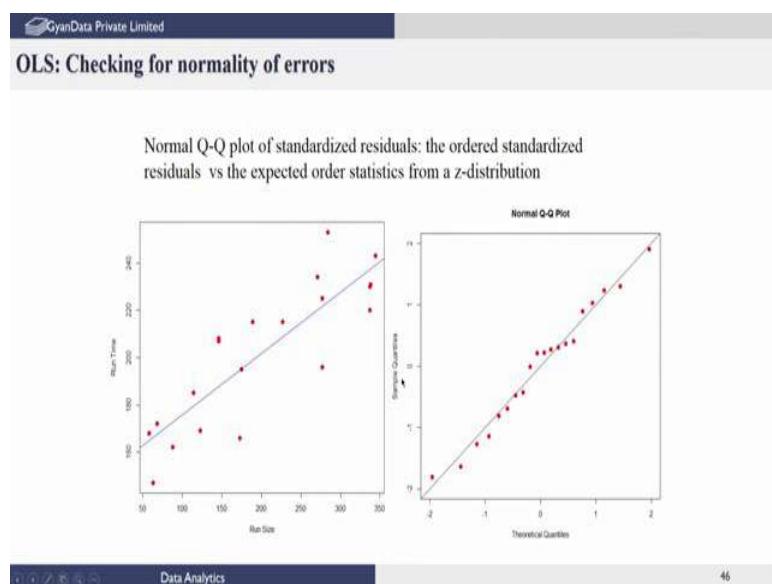
Now, the test for normality can also be done using the residuals. We have already seen the prob even we did statistical analysis the notion of a probability plot. Where we plot the sample quantiles, against the quantiles from the distribution with which we want to compare.

So, if we want to compare, whether a set of given a given set of data follows a certain distribution then we plot the sample quantiles from the quantiles drawn for that particular distribution against which we want to test this sample. So, in this case we want to test, whether residuals that we have the standardized residuals, come from a standard normal distribution. And therefore, we will take the quantiles from the standard normal distribution and plot it. Just to recap what do we mean by quantile it is a percentile data value below which a certain percentage of data lies. For example, if you want to find given a data set, what is the value below which 10 percent of the data lies, maybe -1.28 here we are given, which means 10 percent of the samples lie below -1.28. 20 percent of the samples lie below -0.84 and so on and so forth.

We have computed this, this we can plot against the standard normal values 10 percent value where the probability, between $-\infty$ and the value is 10 percent. And the value between $-\infty$ and that value should be 20 percent and so on so forth. Those represents the x values corresponding to these probabilities and we can use that plot it and then before completing this contents we have arranged the data.

So, we have seen this before I have just only recapped this. And we can use what is called Q-Q norm function in r to actually do it if you give the data set x and ask you to do a probability plot Q-Q norm will do this for you directly in r.

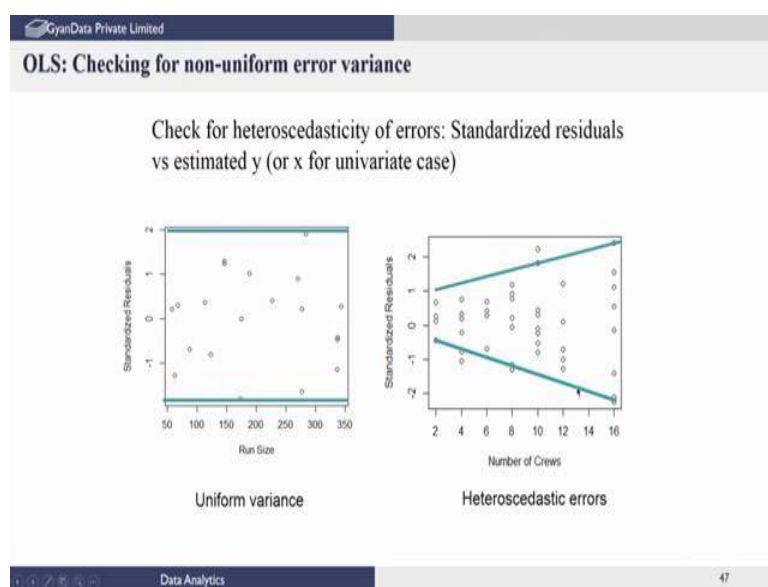
(Refer Slide Time: 17:03)



So, this is a sample Q-Q plot I have taken for some arbitrary random data set samples drawn from the standard normal distribution and you can see that if you do the normal Q-Q plot for the residuals after fitting the regression line, it seems to closely follow the 45-degree line. So, the theoretical quantiles computed from the standard normal distribution and the quantiles computed from the sample residuals, standardized residuals, the fall on the 45-degree line. And therefore, in this case we can safely conclude that the errors in the data come from a standard normal distribution.

So, a Q-Q plot if this thing is in does not happen. If we find the significant deviation of this quantiles from the 45-degree line, then a normal distribution assumption is incorrect. Which means we have to modify the a least square subjective function to actually accommodate this. It may not have it may or may not have a significant effect on the regression coefficient, but there are ways of dealing with it which I will not go into.

(Refer Slide Time: 18:13)



A third thing that we need to test is whether the residual variances I am sorry the error variances in the data are having a uniform variance or I have different variances for different samples, and again here what we do is look at the residual plot and standardized residual versus the predicted values what you have to plot and if you do and look at this thing. It seems to be that the there is no particular trend in the residuals. For example, in the right-hand side we find that the residuals close to when the number of values is - 2 is 2 is spread is very small.

Whereas, when the number of crews is 16 the spread is very high. So, the spread increases or looks like a funnel when we actually look at

the residuals. Whereas, such a effect is not found on the data set corresponding to the left-hand side thing. So, here I have plotted the standardized residual for 2 different data sets. Just to illustrate the type of figures you might get. If you get a figure such as in the left then we can safely conclude that the errors in different measurements have the same variance, whereas if you have a funnel type of effect then you know that the errors, where a variances increases as the value increases. So, it depends on the value itself, which implies that you cannot use a standard least squares method, you should use a weighted least squares method.

So, data points which are corresponding to these 4 should be given more weight and data points corresponding to this should be given less weight and we call that a weighted least squares. That is the way we have to deal with what we call heteroscedastic errors of the sky. Again, I am not going to go into the whole thing I just want to illustrate, that first the residual plots are used in order to verify the assumptions and if the assumptions are not valid then we have correction mechanisms to modify our regression procedure. But linear this does not indicate a linear model is not adequate, the linear model adequacy test is basically based on the pattern if there is no pattern in the residuals you can go ahead and assume that the linear model t is adequate as long as the other measures are also satisfactory. But here it is related to the error variances, and in this case, we only modify the linear regression method, and we still go with a linear model for this for these case such as the one shown on the right.

(Refer Slide Time: 21:01)

GyanData Private Limited

OLS: Checking for outliers in data

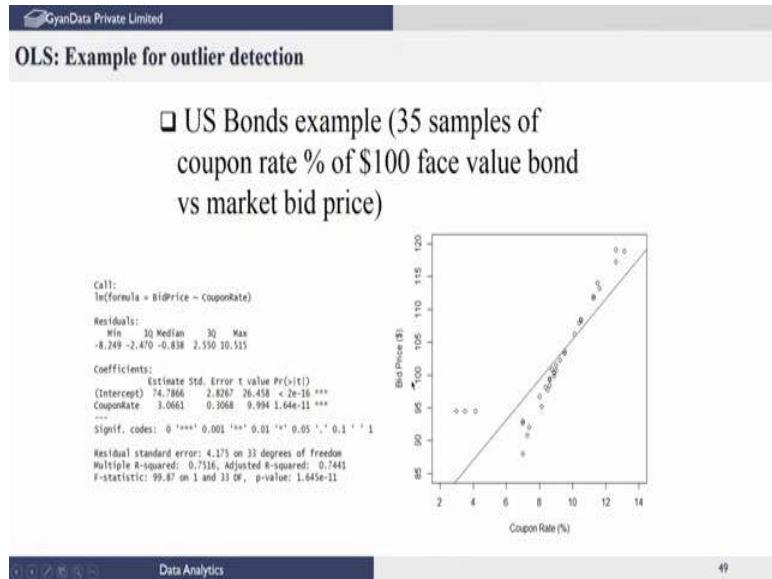
- ❑ Outliers: Points which do not conform to the pattern in bulk of the data
- ❑ Outliers can be identified using hypotheses test of residual of each sample
 - For a 5% level of significance a sample is considered an outlier if the corresponding standardized residuals of lie outside [-2, 2]
 - Even if several residuals lie outside confidence region, identify only one outlier at every iteration (corresponding to the sample with largest standard residual magnitude) – An outlier in a sample can ‘smear’ to other samples due to the regression
 - Apply regression to reduced sample set and iterate until no outliers are detected

The last thing that we need to do is also clean out the data, we do not want to use data that have got large errors what we call outliers, points which do not conform to the pattern that is found in the bulk of the data. And the outliers can be easily identified using hypothesis test of the residual for each sample. We have actually found the residual for example, we have actually found the standardized residuals. So, the standardized residual roughly follow we know it follows a t distribution. But we can for large enough number of samples, we can assume that it follows a normal distribution. So, if I use a 5 percent level of significance we can run a test, hypothesis test for each sample residual, and if the residual lies outside - 2 to + 2, we can conclude that the sample is an outlier.

So, for each sample, we test whether the sample standardized residual lies outside of this interval and if it lies outside this interval, we can conclude that that particular sample maybe an outlier and remove it from our data set. The only thing when we do out lie detection is this, it may turn out that we do that first time we fit a regression, and do an outlier detection we may find several residuals. Lying outside the confidence interval - 2 to + 2 -95 percent confidence interval in which case we do not throw all the samples out at the same time. We only throw out the one that is most offending. Which is we identify the outlier that corresponds to the sample with the largest standard standardized residual magnitude; which of which is furthest away from - 2 or + 2. That is the one we will take and remove it.

Once we remove that we again run a regression on the remaining samples, and again run this outlier detection test. So, we do remove only one outlier at a time. The reason for this is, when we perform an outlier detection we should be aware that a single outlier can smear affect the residuals of other samples because of our regression parameters are obtained from all the data points. Therefore, even a single outlier can cause other outliers to fall outside the confidence interval. Therefore, we do not want hastily conclude that all residuals falling outside the confidence interval are outliers. Only the one that has the maximum magnitude we actually take it out. And then we redo this so, that one at a time we do it will be a safe way of performing outlier detection.

(Refer Slide Time: 23:34)



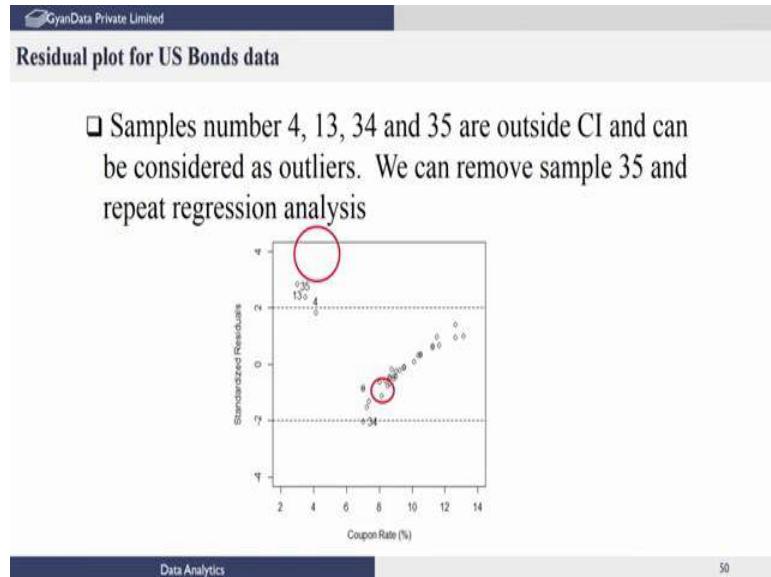
Again, we will illustrate this with an example. Here is us bonds example which consists of 35 samples, us bonds whose face value is 100 dollars, and it is a guaranteed interest rate is provided for each of these bonds depending on when they were released and so on, and that are different bonds with different interest rates. But these are also traded in the market, and the selling price in the market or bid price would be different depending on the kind of interest rate they attract. So, you would presume that the bond which has a higher interest rate would have a higher market price.

So, there might be a linear correlation or linear relation between the market price and the interest rate for that bond. So, here there are 35 samples that are obtained from nothing. These data sets are standard data sets that you can actually download from the net, if you just search for it, you will get it just like the Anscombe data set, and the what you call computer repair time data set that I have been using in the previous lectures. If you perform a regression, and you get a t of this kind. So, it shows that the linear t seems to be adequate you can run the r command lm and you will find that the intercept is 74.8 and the slope is 3.6, and standard errors given and clearly the what you called the p value is very, very low; which means that you will not reject the significance that is the intercept is significant and the slope is also significant, they are not close to 0. You can of course, compute confidence interval and come to the same judgment.

You can run an f test, here also it says the p value of the f test is - 11, which means you will reject the null hypothesis. And conclude that a full model is adequate which means the slope is important here. So, the r value seems to be reasonably good 0.75. And so, we can say the

initial indicators are that a linear model is adequate. Now let us go ahead and do the residual analysis for this.

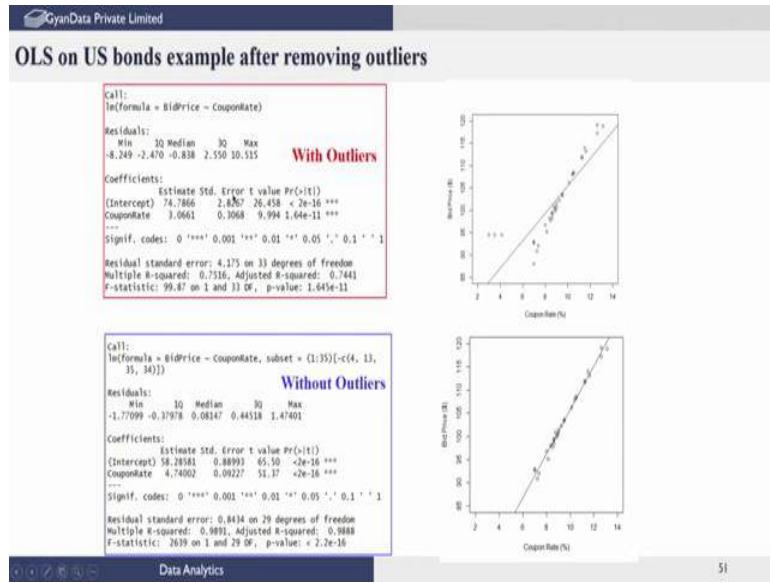
(Refer Slide Time: 25:44)



We perform a residual plot for standardized residual plot, and we find that except for these 4 points, 35 sample number 35 , sample number 13, sample number 34 seems to be outside of the ± 2 confidence interval, and they maybe you may conclude that these are out outliers. While the others are within the bond and they are definitely not out outliers. There seems to be some kind of a pattern, as the coupon that increases the standardized residuals increase. So, maybe as there is a certain amount of non-linearity in the model, but let us remove these outliers before making a final conclusion.

So, we can remove all these 4 outliers at the same time, if you want as I said that is not a good idea perhaps we should remove only sample number 35 which has furthest away from the boundary with the residual with the largest magnitude. And redo this because of lack of time, I have just removed all 4 at the same time and then done the analysis. My suggestion is you do one at a time and then repeat this exercise for yourself. Here we have removed these 4 samples 4 13 35 and then run the regression analysis.

(Refer Slide Time: 27:02)



Again, you can see that the regression analysis, maintain retaining all the samples is shown on the right-hand side of the plot. And their corresponding intercept coupon the slope as well as the f test statistic and so on r squared value is shown here. And once we remove these 4 samples which we outliers and then rerun it, now the fit seems to be much better. It is also seen on the left-hand side that the t is much better you can see that the r squared value has gone up to 0.99 from 0.775. The again the test on the intercept and the coupon rate or slope shows that that they are significant and therefore, you should not assume that they are close to 0.

It also shows that the f statistic has also a low p value which means, you raise the null hypothesis that reduce model is adequate which means the linear model with the slope parameter is a much better fit. So, all of this indicator show seem to indicate show that a linear model is adequate and the t seems to be good. But we should do a residual plot again with this data. And if that actually shows no pattern we can actually stop there we can say that are no outliers. And therefore, we can conclude that regression model that we have fitted for this data is a reasonably good one.

Next class will see how to actually extend all of these ideas to the multiple linear regression which consist of many independent variables and one dependent variable.

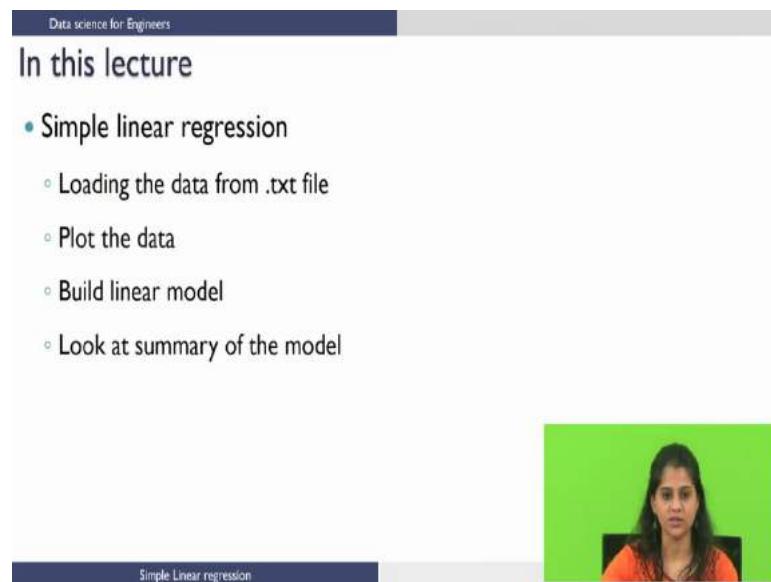
Thank you.

Data Science for Engineers
Prof. Shweta Sridhar
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 35
Simple Linear Regression Modelling

Welcome to the lecture on the implementation of simple linear regression using R

(Refer Slide Time: 00:19)



The image shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. The main title 'In this lecture' is centered above a bulleted list. The list includes: 'Simple linear regression' (with a bullet point), 'Loading the data from .txt file' (with a bullet point), 'Plot the data' (with a bullet point), 'Build linear model' (with a bullet point), and 'Look at summary of the model' (with a bullet point). At the bottom of the slide, there is a dark blue footer bar with the text 'Simple Linear regression'. To the right of the slide, there is a video frame showing a woman with dark hair, wearing an orange top, speaking against a green background.

- Simple linear regression
 - Loading the data from .txt file
 - Plot the data
 - Build linear model
 - Look at summary of the model

In this lecture, we are going to see how to implement simple linear regression in R. As a part of this lecture, we are also going to look at how to load the data from a text file, how to plot the data, how to build a linear regression model and how to interpret the summary of the model?

(Refer Slide Time: 00:39)

Data science for Engineers

Loading data

- Dataset 'bonds' is given in ".txt" format
- To load data from the file the function used is `read.delim()`

Simple Linear regression

Now, let us see how to load the data. Now you have been given the data set bonds in the text format, the extension of the file is dot "txt". To load the data from the file, we use the function read dot delim.

(Refer Slide Time: 00:53)

Data science for Engineers

read.delim()

Reads a file in table format and creates a data frame from it

SYNTAX

```
read.delim(file, row.names=1)
```

file	the name of the file which the data are to be read from. Each row of the table appears as one line of the file.
row.names	a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names.

Simple Linear regression

So, read dot delim reads a file in a table format and creates a data frame out of it. The input arguments to the function are file and row dot names. Now file is the name of the file from which you want to read the data and row dot names are essentially the row ids. It can be a

vector of names or only a single number corresponding to the column name.

(Refer Slide Time: 01:16)

The slide has a dark blue header bar with the text "Data science for Engineers" on the left and a navigation menu on the right. The main content area has a dark blue header "Loading data". Below it is a bulleted list:

- Assuming that bonds.txt is in your current working directory

Below the list is a snippet of R code:

```
bonds <- read.delim("bonds.txt", row.names=1)
```

- The data is saved into a data frame 'bonds'

At the bottom of the slide, there is a small video player window showing a woman speaking. The video player has a dark blue header "Simple Linear regression".

Now, assuming that the data is in your current working directory, the command reads as read dot delim. So, within quotes I have bonds dot txt and I am giving row dot names = 1. Now, once the command is executed, an object of bonds is created, which is a data frame. Now let us see how to view the data. View of bonds will display the data in a tabular format, the snippet below shows the table.

(Refer Slide Time: 01:40)

The slide has a dark blue header bar with the text "Data science for Engineers" on the left and a navigation menu on the right. The main content area has a dark blue header "Viewing data". Below it is a bulleted list:

- `View(bonds)` will display the dataframe in a tabular format

Below the list is a screenshot of a data viewer tool showing a table with four rows and three columns. The columns are labeled "CouponRate" and "BidPrice". The data is as follows:

	CouponRate	BidPrice
1	7.000	92.94
2	9.000	101.44
3	7.000	92.66
4	4.125	94.50

Below the table is another bulleted list:

- `head(bonds)` and `tail(bonds)` will display the first and last six rows from the dataframe

At the bottom of the slide, there is a small video player window showing a woman speaking. The video player has a dark blue header "Simple Linear regression".

We can also view the first few rows of any data set, head and tail functions will help us to do that. Now, head of bonds will give us the first 6 rows from the data and tail of bonds will give us the last 6 rows from the data.

(Refer Slide Time: 02:02)

Data science for Engineers

Description of dataset

- The data has two variables CouponRate and BidPrice.
- CouponRate refers to the fixed interest rate that the issuer pays to the lender.
- BidPrice is the price someone is willing to pay for the bond.



Simple Linear regression

Now, let us look at the description of the dataset, now the data has 2 variables coupon Rate and Bid Price. Now, coupon rate refers to the fixed interest rate that the issuer pays to lender. Bid price is the price someone is willing to pay for the bond.

(Refer Slide Time: 02:19)

Data science for Engineers

Structure of the data

- Each variable and its data type
- `str()`- input is dataframe
- See whether each of the variable datatypes are same as you expect them to be
- If not coerce

```
> str(bonds)
'data.frame': 35 obs. of 2 variables:
 $ CouponRate: num 7 9 7 4.12 13.12 ...
 $ BidPrice  : num 92.9 101.4 92.7 94.5
```



Simple Linear regression

Now, we have seen how to load the data and how to view the data. Let us now see what the structure of the data is. By structure I mean that each variable and its data type. We use the function str and the input to the function is a dataframe. Now we exactly want to see whether the variable data types are same as what we expected them to be, if not we need to coerce them to the respective data types.

So, now this should ring a bell, because we have learned the function as dot followed by the name of the data type and we will use this function to coerce it if the variable is not of the desired type. Now for this dataset, I run the function I say str of bonds now bonds is the name of my data frame. So, the output reads as data frame bonds is of the type data frame it has 35 observations of 2 variables. The first column being coupon rate, which is of the type numeric and I have the first few values being displayed.

The next column Bid Price is also of the type numeric and the first few values of the same column are being displayed.

(Refer Slide Time: 03:32)

Data science for Engineers

Summary of the data

- Gives mean and five number summary

```
> summary(bonds)
   CouponRate      BidPrice
Min. : 3.000  Min. : 88.00
1st Qu.: 8.062 1st Qu.: 95.95
Median : 8.875 Median :100.38
Mean   : 8.921 Mean  :102.14
3rd Qu.:10.438 3rd Qu.:108.11
Max.   :13.125 Max.  :119.06
```

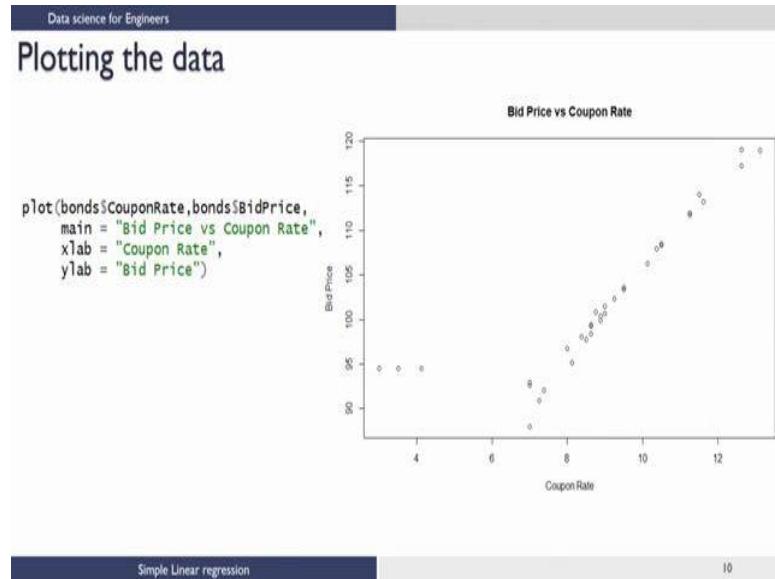


Simple Linear regression

Now, let us look at the summary of the data. So, the summary function followed by the name of the data frame in this case bonds will give us 5 number summary and mean from the data.

Now, the first column which is coupon rate, I have the 5 number summary and the mean and I also have the 5 number summary and the mean for the second column. So, till now we have seen how to load the data, how to view the data, We have also looked at the structure and the summary.

(Refer Slide Time: 04:02)



Now, let us see how to visualize the data. So, to visualize the data I use the plot function. We have covered the plot function earlier in the visualization in r section, now the input to the plot function are basically x and y. In this case x refers to my coupon rate and y refers to my bid price. So, in order to access the variables, I need to give the name of the data frame followed by a dollar symbol. So, I say access coupon rate from the bonds data and access bid price from the bonds data.

So, I can also give a title to my plot. So, inside the parameter main you can specify the title of your plot, xlab is nothing, but x label. So, I am assigning it as x "Coupon Rate" and y label I am assigning it as "Bid Price". So, the plot is on right hand side.

So, the title is bid price versus coupon rate like how we have assigned it on the y axis I have bid price and I have labeled it as bid price and on the x axis, I have coupon rate and I have labeled it as coupon rate.

Now, we see a linear trend. Now there are some points which are completely outside the range of coupon rate. Now let us see if our linear model will help us to identify these points.

(Refer Slide Time: 05:19)

Data science for Engineers

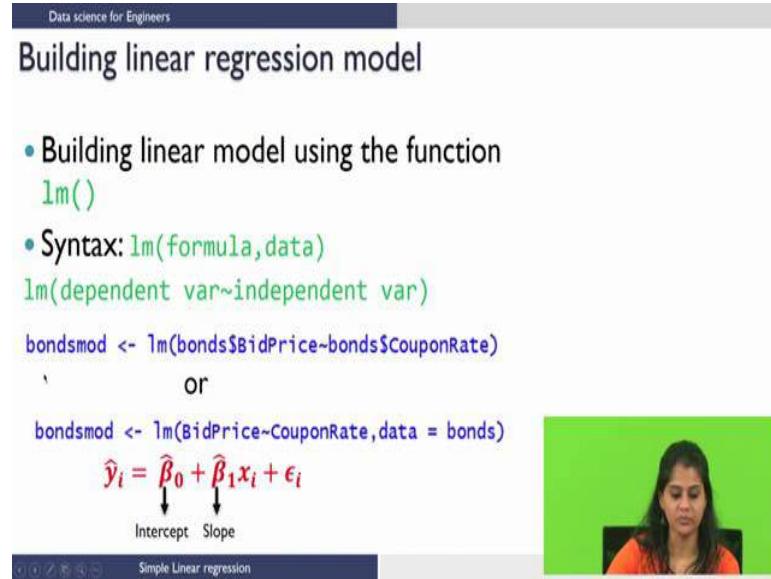
Building linear regression model

- Building linear model using the function `lm()`
- Syntax: `lm(formula,data)`
`lm(dependent var~independent var)`

```
bondsmod <- lm(BidPrice~CouponRate,bonds)
or
bondsmod <- lm(BidPrice~CouponRate,data = bonds)
```

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \epsilon_i$$

↓ ↓
Intercept Slope



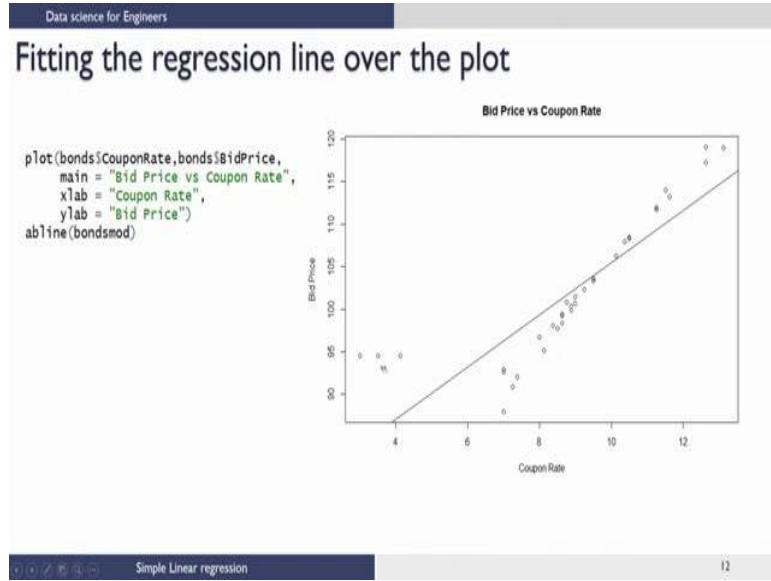
Simple Linear regression

So, to start with let us build a linear regression model. So, building a linear model is done using the function `lm`. So, the inputs for the function are formula and data, by formula I mean I am regressing dependent variable versus the independent variable.

So, how is it translated to a formula? So, I have dependent variable I have a tilde sign followed by the independent variable. So, the tilde sign tells us regress the dependent variable with the independent variable. So, there are 2 ways to build the linear model, let us see how to do that. The first way tells us linear model which is a function. So, I am accessing the individual variables which is bid price and coupon rate using the data frame followed by the dollar symbol. So, take bid price from the bonds data and take coupon rate from the bonds data and regress them.

There is also another way. So, instead of mentioning the name of the data frame to access the variables, we can directly mention the name of the variable and give `data = bonds`. So, access these variables from the bonds data. So, assuming our equation is of the form $\hat{y}_i = \beta_0 + \beta_1 x_i + \epsilon_i$. So, ϵ_i is the error which will be called as residuals. So, β_0 is the intercept and β_1 is the slope. So, hereafter these estimates will be referred to as intercept and slope. So, now that we have built a linear model and have saved it as an object `bondsmod` let us see how to fit the regression line over the plot.

(Refer Slide Time: 07:02)

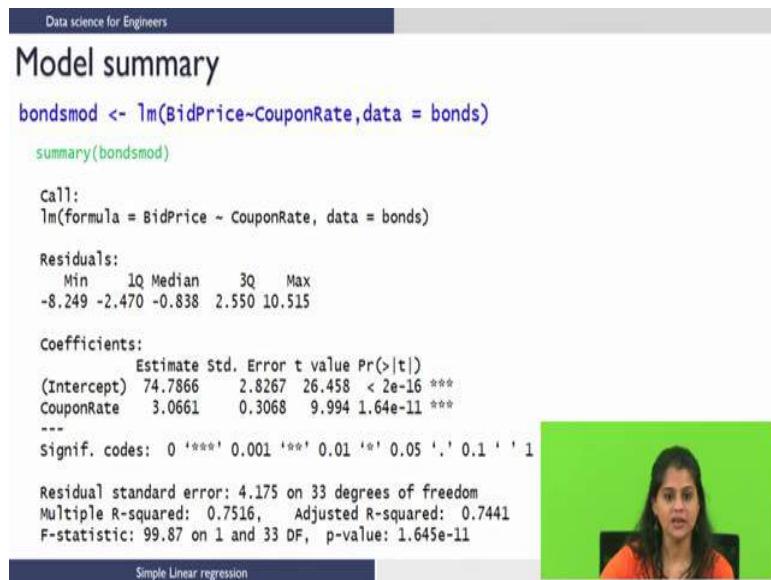


We will use a function called ab line and the input for the function is bondsmod which is my linear model.

Now, we have already gone back and seen how to plot coupon rate and bid price. Now in addition to the plot you need to mention this command. So, ab here refers to the intercept and slope. If your equation is of the form $y = a + bx$, then a is my intercept and b is my slope.

In this case a is β_0 and b is β_1 . So, let us see how the plot looks. So, on my right I have the plot we are now able to see how the regression line fits. It fits pretty badly and it is also not identify the outliers. So, we can say that regression line is indeed getting affected by these outliers.

(Refer Slide Time: 07:59)



So, now let us take a look at the model summary. So, I have regress bid price versus coupon rate from the data bonds, you can also use the other command.

So, summary is a function the input to the summary function becomes a linear model. So, we have bondsmod as the linear model, this is the first look at the summary. So, this is how it looks when you run the command and this is how it would look in the callzone.

So, we have 4 sections of output we have call, we have residual, we have coefficients, and we have some few heuristics at the bottom. So, now, let us look at each of these and what they mean in depth.

(Refer Slide Time: 08:37)

```

Call:
lm(formula = BidPrice ~ CouponRate, data = bonds)

Residuals:   Min     1Q Median     3Q    Max 
-8.249 -2.470 -0.838  2.550 10.515 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 74.7866  2.8267 26.458 < 2e-16 ***
CouponRate   3.0661  0.3068  9.994 1.64e-11 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.175 on 33 degrees of freedom
Multiple R-squared:  0.7516, Adjusted R-squared:  0.7441 
F-statistic: 99.87 on 1 and 33 DF, p-value: 1.645e-11

```

Simple Linear regression

14

Now, call displays the formula which we have used. So, in this case I have used the formula bidprice versus coupon rate so regress bidprice, which is my dependent variable with my independent variable which is coupon rate and from the data bonds.

Now, this is the way for you to check if you have given the right dependent and independent variable. The next section is residual. So, what are residuals are nothing, but difference between the observed and predicted values. So, in our equation earlier we saw we had a parameter called ϵ_i . So, that ϵ_i corresponds to residuals below the residuals is the 5 number summary for the residual.

So, the next section is coefficients. We see 2 rows which is intercept and coupon rate and certain set of values associated with them. Now these intercept and coupon rate are nothing, but β_0 and β_1 hat, we earlier saw for our equation $y = \beta_0 + \beta_1 x_i + \epsilon$, β_0 is the intercept and β_1 is the slope.

Now, let us see what other 4 parameters in the column have to say. Now I have the first column which is estimate. Now this is nothing, but the estimate for the slope and intercept parameter. The next column is

standard error. So, standard error is the estimated standard deviation associated for the slope and intercept.

(Refer Slide Time: 10:08)

```

Data science for Engineers
Model summary

call:
lm(formula = BidPrice ~ CouponRate, data = bonds)

Residuals:
    Min      1Q Median      3Q     Max 
 -8.249 -2.470 -0.838  2.550 10.515 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 74.7866   2.8267 26.458 < 2e-16 ***
CouponRate  3.0661   0.3068  9.994 1.64e-11 *** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.175 on 33 degrees of freedom
Multiple R-squared:  0.7516, Adjusted R-squared:  0.7441 
F-statistic: 99.87 on 1 and 33 DF, p-value: 1.645e-11

Probability of seeing a 't' random variable will be greater than the observed t-statistic
Significance level above which the null hypothesis will be rejected
Ratio of estimate by the standard error. Used as a criterion for hypothesis testing
F statistic to test whether slope=0

```

Simple Linear regression

15

I have the next column as t value. So, what is t value? It is the ratio of estimate by the standard error and it is also an important criterion for the hypothesis testing. The column after that is the probability.

So, it is the probability of seeing a t random variable, which will be greater than the observed t statistic. So, we can see few stars being indicated at the end. So, what are these stars. These stars tell us the significance level above, which the null hypothesis will be rejected.

So, what is the null hypothesis? The null hypothesis is that the estimates will be = 0. At the last line I have an F statistic and a corresponding p-value associated with it. Now the F statistic is again used to test the null hypothesis which is nothing, but slope = 0.

So, in this lecture we saw how to load a data, how to plot and how to visualize, how to build a linear model and how to interpret the results from the linear model? So, in the next lecture we will see how to assess our model and we will see if we can improvise our model.

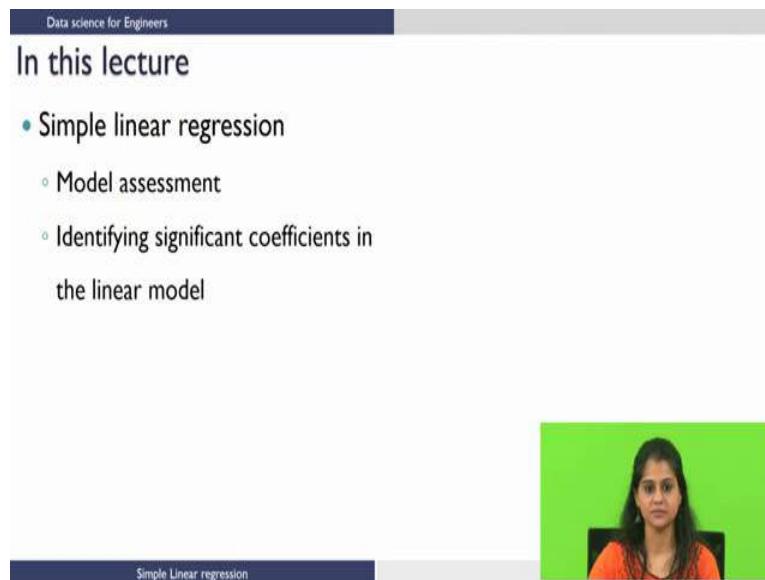
Thank you.

Data Science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 36
Simple Linear Regression Model Assessment

So, welcome to the second lecture on implementation of simple linear regression using R. In the last lecture, we saw how to read a data from a text file, how to visualize the data, how to build a linear model, and how to interpret it.

(Refer Slide Time: 00:31)



The image shows a presentation slide with a dark blue header bar containing the text "Data science for Engineers". The main content area has a light gray background and features the title "In this lecture" in bold black font. Below the title is a bulleted list of topics:

- Simple linear regression
 - Model assessment
 - Identifying significant coefficients in the linear model

At the bottom of the slide, there is a dark blue footer bar with the text "Simple Linear regression".

Below the slide, there is a video frame showing a woman with dark hair, wearing an orange top, sitting in front of a green screen. She appears to be speaking or presenting.

In this lecture, we are going to look at simple linear regression model assessment. As a part of this, we are also going to look at how to identifying, significant coefficients in the linear model.

(Refer Slide Time: 00:40)

- How good is the linear model?
- Which coefficients of the linear model are significant (Identify important variables)
- Can we improve quality of linear model?
 - Are there bad measurements in the data (outliers)



Simple Linear regression

Now, let us start with model assessment. So, there are a few questions which we need to answer before we go into model assessment. After having built a model, we first need to check how good is our linear model. Now, we need to identify which coefficients in the linear model are significant.

Now, if you have multiple independent variables, then we also need to identify which of them are important. We also need to know can we improvise the model further. As a part of this, we are going to look at are there any bad measurements in the data. So, by bad measurements we mean are there any other outliers in the data which could affect the model. This question alone will be handled in the next lecture.

So, let us look at how to answer the first two questions.

(Refer Slide Time: 01:23)



Data science for Engineers

Model summary

```
Call:
lm(formula = BidPrice ~ CouponRate, data = bonds)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.249 -2.470 -0.838  2.550 10.515 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 74.7866    2.8267 26.458 < 2e-16 ***
CouponRate   3.0661    0.3068  9.994 1.64e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.175 on 33 degrees of freedom
Multiple R-squared:  0.7516, Adjusted R-squared:  0.7441 
F-statistic: 99.87 on 1 and 33 DF,  p-value: 1.645e-11
```

Simple Linear regression

Now, from the earlier lecture we saw how to look at the summary. We also know how to interpret it now. Now, this is the first gist of summary that you get when you run the command.

So, I had regressed BidPrice with coupon rate from the data bonds and bondsmod was my linear model. I also have the estimates here which are nothing but the intercept and slope. So, let us look at the first level of model assessment.

(Refer Slide Time: 01:45)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header, the main title 'First level model assessment' is displayed in a large, bold, black font. Underneath the title, there is a bulleted list of topics: '

- R² value=0.7516
- Hypothesis Testing
 - Coefficients
 - Full model and reduced model

'. At the bottom of the slide, there is a small video frame showing a woman with long dark hair, wearing an orange top, speaking against a green background. Below this video frame, a dark blue footer bar contains the text 'Simple Linear regression'.

So, the first level of model assessment is done using the R squared value. Now if you go back and see, the R squared value for our model is 0.7516. Now this is pretty close to 1. Though not very close, but it is still closer to 1.

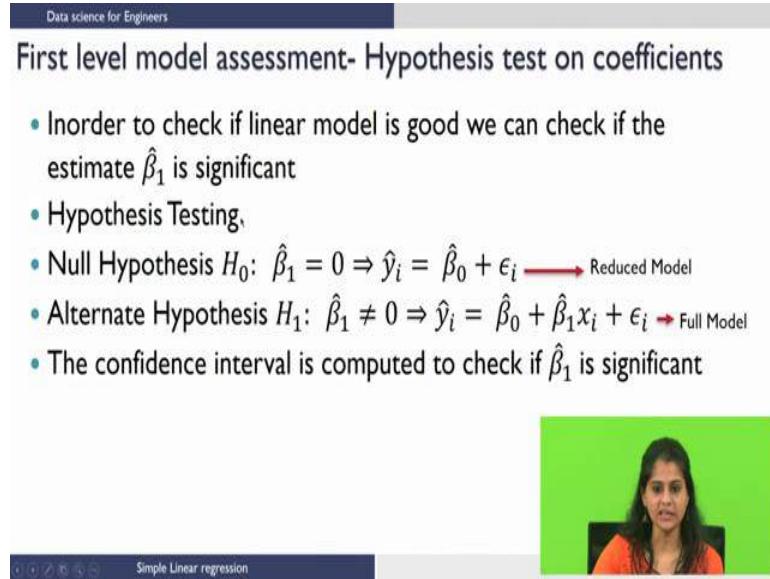
So, we can say that, yes, the model we have developed is reasonably good, but not really good. It also tells us the assumption that we made initially to begin with, that there is a linear relationship between x and y. We are also going to look at hypothesis testing. As a part of this, we are going to look at the hypothesis testing on coefficients and then on the full and reduced model. Now, let us see what these full and reduced model are. So, first let us do the hypothesis test on coefficients.

(Refer Slide Time: 02:36)

Data science for Engineers

First level model assessment- Hypothesis test on coefficients

- Inorder to check if linear model is good we can check if the estimate $\hat{\beta}_1$ is significant
- Hypothesis Testing,
- Null Hypothesis $H_0: \hat{\beta}_1 = 0 \Rightarrow \hat{y}_i = \hat{\beta}_0 + \epsilon_i$ Reduced Model
- Alternate Hypothesis $H_1: \hat{\beta}_1 \neq 0 \Rightarrow \hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \epsilon_i$ Full Model
- The confidence interval is computed to check if $\hat{\beta}_1$ is significant



So, in order to check, if your linear model is good. We can check if the estimate β_1 which is the slope, whether it is significant. So, my null hypothesis is, β_1 which is the slope = 0.

So, this means that \hat{y}_i which is my predicted value = β_0 , which is the intercept + ϵ_i . We also learnt in the earlier lecture, that this ϵ_i is the residual. Now this becomes my reduced model.

Since, my slope = 0. So, what will the alternate hypothesis be in this case? So, my alternate hypothesis is that, $\beta_1 \neq 0$, and my \hat{y}_i which is the predicted value = $\beta_0 + \beta_1 x_i + \epsilon_i$.

(Refer Slide Time: 03:36)

Data science for Engineers

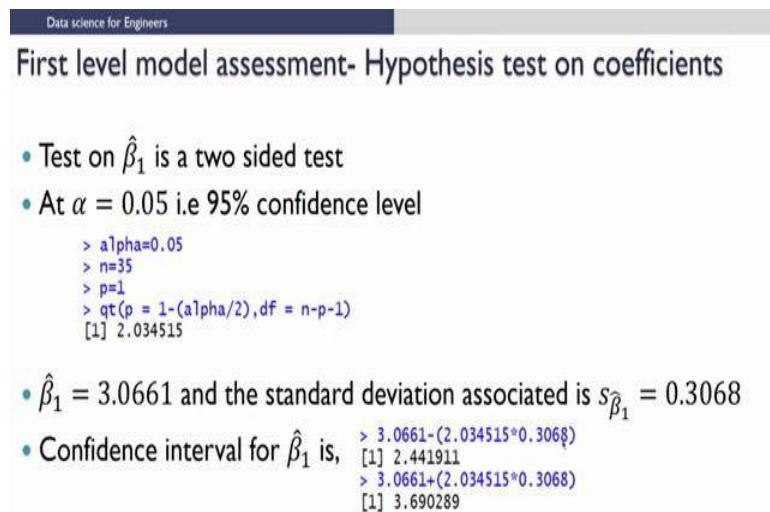
First level model assessment- Hypothesis test on coefficients

- Test on $\hat{\beta}_1$ is a two sided test
- At $\alpha = 0.05$ i.e 95% confidence level

```
> alpha=0.05
> n=35
> p=1
> qt(p = 1-(alpha/2),df = n-p-1)
[1] 2.034515
```

- $\hat{\beta}_1 = 3.0661$ and the standard deviation associated is $s_{\hat{\beta}_1} = 0.3068$
- Confidence interval for $\hat{\beta}_1$ is,

```
> 3.0661-(2.034515*0.3068)
[1] 2.441911
> 3.0661+(2.034515*0.3068)
[1] 3.690289
```



Now, this becomes my full model. The confidence interval is computed to check if the slope is significant. Now, this test is a two sided test, since we see $\beta_1 = 0$ or $\neq 0$. At 95 percent confidence level, that is, at $\alpha = 5$ percent. We get the critical value to be 2.0345. Now, let us see how to compute this critical value.

So, we know that $\alpha = 0.05$. And n here is the number of observations in your data. Now, in this case I have 35 observations. P becomes my number of independent variables. Here I have only my one independent variable. So now, we know from the statistics module how to compute the quantiles for a t from a t distribution.

Now, I give $p = 1 - \alpha$ by 2 since it is a two sided test, and the number of degrees of freedom are given as $n - p - 1$. So, this command is in built in R you just need to give the inputs. You need to supply p and degrees of freedom.

After having done this we get the quantiles to be = 2.03. So, this is the critical value we are going to use this to compute the confidence interval. Now, we earlier saw from the summary that the slope, which is nothing but the $\beta_1 = 3.0661$ and the standard deviation associated with it was 0.3068. So, the confidence interval is computed as the estimate + or - the critical value into the standard error.

So, by doing so we get the lower bound as 2.44, and the upper bound as 3.69. So now, we know that, this interval does not encompass 0, that is, anywhere between the interval I do not have 0. So, this itself is indicative of the fact that my β_1 that is the slope is significant.

(Refer Slide Time: 05:36)

Data science for Engineers

First level model assessment- Hypothesis Test on models

- Computing F statistic

$$F_o = \frac{SST-SSE}{SSE/(n-2)} = \frac{SSR}{SSE/(n-2)}$$

SSE = $\sum(y_i - \hat{y}_i)^2$
SSR = $\sum(\hat{y}_i - \bar{y})^2$

```

> SSE<-sum((bonds$BidPrice-bondsmod$fitted.values)^2)
> SSE
[1] 575.3418
> SSR<-sum((bondsmod$fitted.values-mean(bonds$BidPrice))^2)
> SSR
[1] 1741.263
> n=35
> (SSR/SSE)^(n-2)
[1] 99.87401

```

- This F statistic is returned by the summary command

Now, let us do a hypothesis test on the models. So, to do so we use the F statistic. So, let us go back and revisit what the F statistic is. So, F statistic is nothing but my sum squared residual divided by the sum square error by the degrees of freedom for the denominator. So, for the sum square residual, we know it is of the form of summation of $\hat{y}_i - \bar{y}$ the whole square. So, I have only one degrees of freedom, since, I am using only one parameter to compute it.

Whereas for the sum square error it is the summation of $(y_i - \hat{y}_i)^2$. Now, I am using two parameters to compute it. So, the degrees of freedom reduced by 2. So, hence I have the denominator as $n - 2$. So, this is how you would compute the sum squared error. So, I am summing my y_i which is nothing but from bid price bond dollar BidPrice. I have the fitted values, and I know the mean which is \bar{y} of the bid price. I am squaring the term, and I am summing it. Now, we know that my num the number of observations we have a 35 from the data. So, from the formulae we know our F statistic is computed as SSR by SSE into $n - 2$. Degrees of freedom $n - 2$ go to the numerator, and we get the F statistic to be = 99.87.

Now, this F statistic is what is returned by the summary, which is given in the last line of the summary.

(Refer Slide Time: 07:16)

Data science for Engineers

First level model assessment- Hypothesis Test on models

- The F statistic from table for 1 and 33 degrees of freedom is 4.17 at 5% significance level
- The observed value of F statistic is 99.87 which is greater than the theoretical



Simple Linear regression

So, let us see what conclusions can we draw from these two tests. We know that the F statistic from the table. 1 and 3 degrees of freedom is 4.17 at 5 percent significance level.

What we observe is 99.87, at 1 and 33 degrees of freedom. Now, this is greater than the theoretical value that we get from the distribution.

(Refer Slide Time: 07:40)

Data science for Engineers

First level model assessment- Hypothesis test on coefficients

- Conclusion:
- Reject the null hypothesis since the confidence interval does not include 0
- Therefore $\hat{\beta}_1$ is significant



Simple Linear regression

So, what conclusion can we draw now? So, we know that, we can reject the null hypothesis, since the confidence interval does not include 0. And hence β_1 which is the slope is also significant.

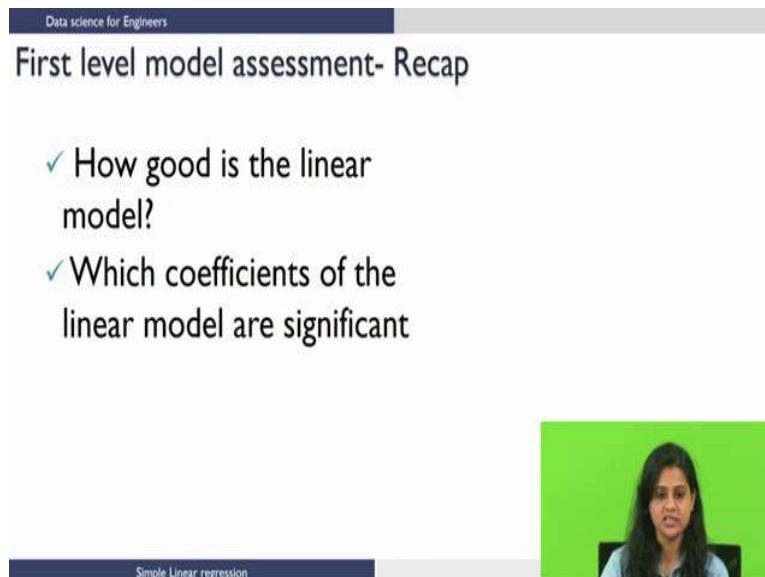
So, in this lecture, we saw how to assess the model. We also looked at how to answer some of the important question that gets associated while assessing a model. We also saw how to identify the significant coefficients. In the next lecture we will look at how to identify outliers and how to improvise a model.

Thank you.

Data Science for Engineers
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture-37
Simple Linear Regression Model Assessment

(Refer Slide Time: 00:21)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header, the main title 'First level model assessment- Recap' is displayed in a large, bold, black font. To the right of the title, there is a video frame showing a woman with long dark hair speaking. At the bottom of the slide, there is a dark blue footer bar with the text 'Simple Linear regression'.

- ✓ How good is the linear model?
- ✓ Which coefficients of the linear model are significant

Welcome to the third lecture on implementation of simple linear regression using R. In the last lecture, we looked at the first level of model assessment, we saw how good is a linear model that we built and we also saw, how to identify the significant coefficients in the linear model.

(Refer Slide Time: 00:35)

- Second level model assessment
 - Can we improve quality of linear model?
 - Are there bad measurements in the data (outliers)



In this lecture, we are going to look at the second level of model assessment. As a part of this, we are going to see, if we can improvise the quality of the linear model and can we identify bad measurements and by bad measurements, we mean outliers.

(Refer Slide Time: 00:52)

Data science for Engineers

Checking for outliers in data

- Outliers: Points which do not conform to the pattern in bulk of the data
- A point is considered an outlier if the corresponding standardized residuals lies outside [-2, 2] at 5 % level of significance

A video frame showing the same woman from the previous slide, now looking directly at the camera. The video frame has a dark blue header bar with the text "Simple Linear regression" in white.

So, let us see, what outliers are. So, outliers are points, which do not con-form to the bulk of the data. Now, a point is considered an outlier, if the corresponding standardized residual falls outside, - 2 and + 2 at 5 per-cent significance level.

(Refer Slide Time: 01:10)

Data science for Engineers

Handling outliers in data

- Even if several residuals lie outside confidence region, identify only one outlier at every iteration
- Apply regression to reduced sample set
- Iterate until no outliers are detected

Simple Linear regression



Now, let us see how to handle these outliers, even if we have several outliers which lie outside the confidence region, we are going to identify only one at a time, at every iteration and after doing so, we are going to apply a linear model on the reduced sample. Now, we are going to iterate, till we detect no more outliers. Now, let us see how to handle these outliers. We are going to start with the residual analysis.

(Refer Slide Time: 01:36)

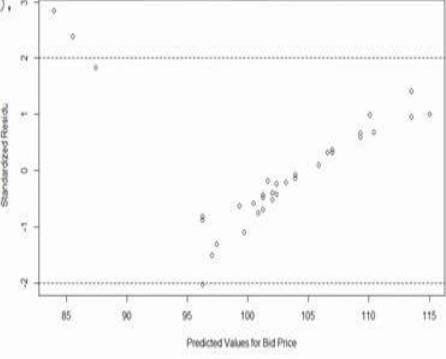
Data science for Engineers

Residual analysis

```
plot(bondsmod$fitted.values,rstandard(bondsmod),  
      main = "Residual plot",  
      xlab = "Predicted Values for Bid Price",  
      ylab = "Standardized Residuals")  
abline(h=2,lty=2)  
abline(h=-2,lty=2)
```

- To know the indices of the outliers we use the function `identify()`

Residual Plot



Simple Linear regression

So, you can see a plot function on the left hand side. Now, for the residual plot, I am going to plot fitted values on the x axis and the standardized residual from the model we have built. Now, we built a

linear model called bondsmod and we are going to calculate the standardized residuals for it.

So, that becomes my y. I am also giving a title like I earlier said. The title for this plot is residual plot and my x label is nothing, but predicted values for bid price. Similarly, my y label is standardized residual. Now, after doing so, we need to set the confidence region. So, let us see how to do that. We again use the same command a b line. Now, a b line is what we have used to fit the linear model onto the plot.

Now, with the same command, we can give the confidence region as well. Now, I am going to set the height, which = h here, as 2 and the line type as 2. So, height is at which you want the line to be drawn and line type is nothing but how you want the line to be drawn. So, you can have dashed lines solid line, dashed and a dot. So, you have several options in there similarly, I also need a lower confidence limit. So, I am setting that to be = - 2 and for the same limit I am setting the line type to be = 2.

Now, let us see how the plot looks. On the right hand side, I have the plot. So, we can see that there are two lines drawn at + and - 2 that defines the confidence level. Now, on the y axis, I have standardized residuals and on the x axis, I have predicted values for bid price. Now, from the plot, we can see that there are two outliers, which are really farther, there is one, which is close to the upper confidence limit. And there is one, which is exactly almost close to the lower confidence limit.

So, let us see how to identify these. So, from the plot, we may not be able to tell which points are these. By points, I mean in the row IDs. We are going to use another function called identify, that will help us identify the indices of these samples. Now, let us see what identify function does.

(Refer Slide Time: 03:53)

Data science for Engineers

identify()

- Reads the position of the graphics pointer when the mouse button is pressed.
- It then searches the coordinates given in x and y for the point closest to the pointer
- If this point is close enough to the pointer, its index will be returned as part of the value of the call

SYNTAX `identify(x,y)`

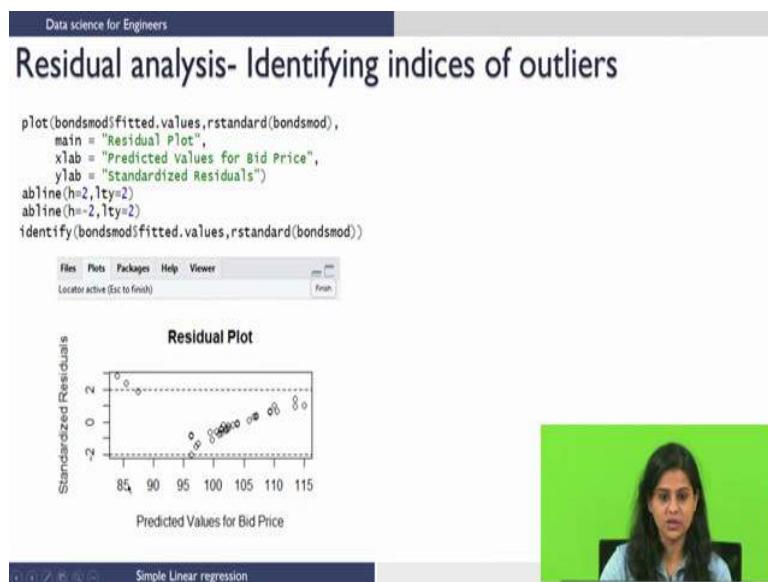
x, y coordinates of points in a scatter plot.

Simple Linear regression

So, it treats the position of the graphic pointer, when the mouse button is pressed it, then searches the coordinates given an x and y for the point closest to the pointer. Now, if the point is close enough to the pointer, its index will be returned as a part of the value code.

Now, let us look at the syntax for it. So, identify is a function and x and y are my input parameters. So, what are my x and y, they are the coordinates of the points in the scatter plot. Now, let us see how to use this function to identify the indices.

(Refer Slide Time: 04:27)



On my left, I have the same commands for the residual plot. So, this is what we saw in the last slide. I now, use the identify function to identify the indices. Now, my input for this is fitted values of the bonds model and the standardized residuals the reason. I am giving fitted values is, because on the plot, I want the indices to be found. So, the plot has fitted values from the model and the standardized residuals from the model.

So, on this plot I want my indices to be identified. So, I give the same inputs that I have used for the plot command for identify function. So, again here, if you see I have fitted values from the bonds model and I am plotting for the y parameter, I am plotting the standardized residuals. Now, once you execute the command, you will not get the output immediately. What will be displayed is the following snippet on the left, you will see a finish button and you will see a message being displayed. Now, on this plot, we will need to click and identify each of the points. Now, let us see how to do that.

(Refer Slide Time: 05:33)

The screenshot shows the RStudio interface with the following elements:

- Top Bar:** Data science for Engineers
- Console:** Residual analysis- Identifying indices of outliers
- Code Editor:** identify(bondsmod\$fitted.values,rstandard(bondsmod))
 - Clicking near a point adds it to the list of identified points
 - Points can be identified only once
 - If the point has already been identified the following message is printed immediately on the R console
 - > identify(bondsmod\$fitted.values,rstandard(bondsmod))
 - warning: nearest point already identified
 - If the click is not near any of the points then following message is displayed
 - > identify(bondsmod\$fitted.values,rstandard(bondsmod))
 - warning: no point within 0.25 inches
- Plots:** Residual Plot (Standardized Residuals vs Predicted Values for Bid Price)
- Bottom Bar:** Simple Linear regression 36

So, I am displaying the command above to remind, you of the fact that we are using fitted values and standardized residuals to identify. Now, click it near a point, adds it to the list of the identified points. Now, if I am going to click near this point. It is going to identify this point and store it. Now, all these points can be identified only once. Now, if a point has already been identified and you still click near it, then you will get the following message. It will be a warning, which reads as nearest point already identified.

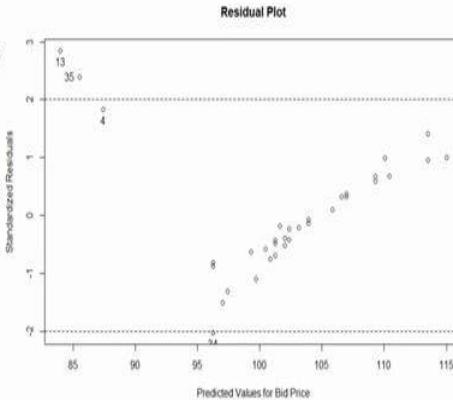
Now, if you do not click near any of the points, then a message is displayed, which says that no point is identified within 0.25 inches. So, if I click here, then I do not have any points closest to it. So, it will display a message saying no point within 0.25 inches.

(Refer Slide Time: 06:29)

Residual analysis- Identifying indices of outliers

- The identification process is terminated by clicking 'Finish'

- After terminating, the indices are displayed on the console and on the plot
- ```
> identify(bondsmod$fitted.values,
+ rstandard(bondsmod))
[1] 4 13 34 35
```



17

Now, once

you have identified all the outliers, you need to click the finish button that is present on the top right, corner of the graphical window, you can also press escape to finish. Now, after terminating the indices are displayed on the console and on the plot. Now So, you can see on the console, I have the indices being displayed as 4 13 34 35, but this will give you only the value.

So, now, to know where your outliers lie on the plot, I am going to look at the plot. So now, I know the 13th point of the sample is the farthest, which is here, followed by the 35th sample, followed by the sample 4 and then I have one more sample, which is here, which is the 34th sample. So, after identifying these outliers, we are going to start by removing one at a time and we are going to build a new model. Now, let us see how to do that. Now, I will start by removing one point at a time.

(Refer Slide Time: 07:32)

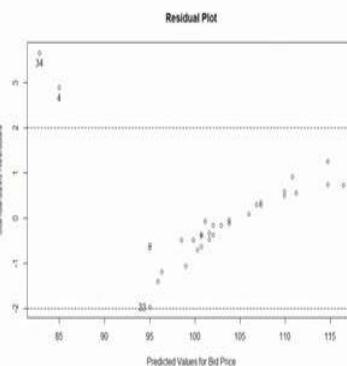
## Removing outliers

- Lets start by removing the farthest outlier i.e. sample 13 and building a new model

```
bonds_new<-bonds[-13,]
bondsmod1<-lm(bonds_new$BidPrice~
bonds_new$CouponRate)
```

- Identify the indices of the outliers on the residual plot

```
> identify(bondsmod1$fitted.values,
+ rstandard(bondsmod1))
[1] 4 33 34
```



18

The first point that I am going to remove is sample 13 that is the 13th point, because it is the farthest in the plot. So, to start with, I am going to create a new data frame called bonds new and it will have all rows of bonds except the 13th row. So, then I am going to create another object called bonds mod one, which is the linear model that is being built for the new data. So, I am going to regress bid price from the new data frame bonds new with coupon rate from the same data set. Now, after building the new linear model, which does not contain the 13th point, that is an outlier.

We are going to repeat the same process again that is on the residual plot, we are going to identify the outliers for the new data. So, on my right. I already have the residual plot with the outliers being identified. So, from the snippet, we can see that for the new data, I have my 4th point, 33rd point and 34th point being, are being identified as outliers. So, now, this new data will contain only 34 data observations, because we have already removed one observation. So, the indices for the new data will change.

So, the farthest point in this data is the 34th point and after that I have the 4th point and followed by that, there is also one point on the line, which is the 33rd point, for this new data. Now, we can see that, if you compare this plot and the earlier plot this point, which is located here was below this line and that is because we had an extreme outlier in the previous case, that had a smearing effect on the remaining points. Now, after building this new linear model let us take a look at the summary.

(Refer Slide Time: 09:21)

```

Data science for Engineers
Comparison between old and new model

With outliers
> summary(bondsmod)
Call:
lm(formula = BidPrice ~ CouponRate, data = bonds)
Residuals:
 Min 1Q Median 3Q Max
-8.249 -2.470 -0.838 2.550 10.515
Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 74.7866 2.8267 26.458 < 2e-16 ***
CouponRate 3.0661 0.3068 9.994 1.64e-11 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
Residual standard error: 4.175 on 33 degrees of freedom
Multiple R-squared: 0.7516, Adjusted R-squared: 0.7441
F-statistic: 99.87 on 1 and 33 DF, p-value: 1.645e-11

Without sample 13
> summary(bondsmod1)
Call:
lm(formula = bonds_new$BidPrice ~ bonds_new$CouponRate)
Residuals:
 Min 1Q Median 3Q Max
-7.0393 -1.7780 -0.5931 1.6511 11.7264
Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 70.5679 2.8147 25.07 < 2e-16 ***
bonds_new$CouponRate 3.4959 0.3016 11.59 5.42e-13
(Intercept) ***
bonds_new$CouponRate ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
Residual standard error: 3.683 on 32 degrees of freedom
Multiple R-squared: 0.8077, Adjusted R-squared: 0.8017
F-statistic: 134.4 on 1 and 32 DF, p-value: 5.417e-13

```

On the left is the summary of the old model bondsmod that contains all the points. On the right, I have the summary of the new model, which does not contain the 13th sample. So, from the R squared values of the two model, we can see that there is a drastic change by just removing one extreme point. So, from 0.17516, the R square improves to 0.8077. So, that is a quite drastic change. Now, let us remove all the other points one by one and let us see how the R squared value changes.

(Refer Slide Time: 10:00)

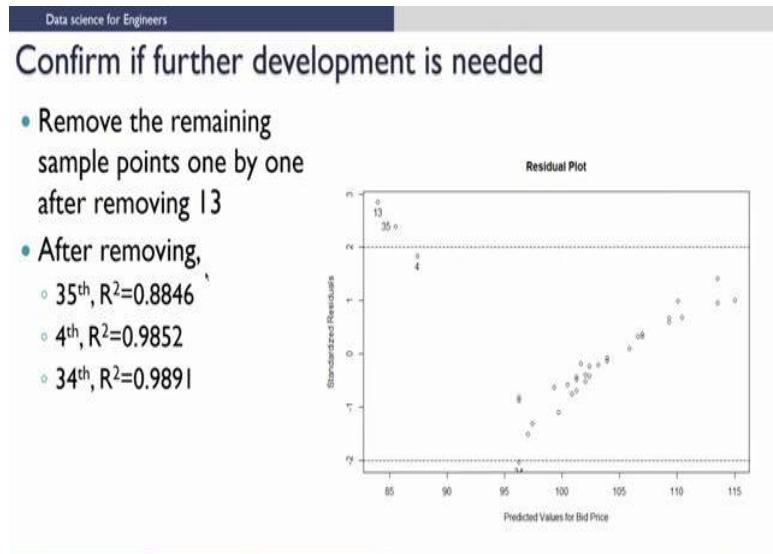
Data science for Engineers

## Comparison between old and new model

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>With outliers</b></p> <pre>&gt; summary(bondsmod) Call: lm(formula = BidPrice ~ CouponRate, data = bonds)  Residuals:     Min      1Q Median      3Q     Max  -8.249 -2.470 -0.838  2.550 10.515   Coefficients:             Estimate Std. Error t value Pr(&gt; t )     (Intercept) 74.7866   2.8267 26.458 &lt; 2e-16 *** CouponRate  3.0661   0.3068 9.994 1.64e-11 ***   ... Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '  Residual standard error: 4.175 on 33 degrees of freedom Multiple R-squared:  0.7516, Adjusted R-squared:  0.7441  F-statistic: 99.87 on 1 and 33 DF, p-value: 1.645e-11</pre> | <p><b>Without sample 13</b></p> <pre>&gt; summary(bondsmod1) Call: lm(formula = bonds_new\$BidPrice ~ bonds_new\$CouponRate)  Residuals:     Min      1Q Median      3Q     Max  -7.093 -1.7700 -0.5931  1.6511 11.7264   Coefficients:             Estimate Std. Error t value Pr(&gt; t )     (Intercept) 70.5679   2.8147 25.07 &lt; 2e-16 *** bonds_new\$CouponRate 3.4959   0.3016 11.59 5.42e-13  ... Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '  Residual standard error: 3.683 on 32 degrees of freedom Multiple R-squared:  0.8077, Adjusted R-squared:  0.8017  F-statistic: 134.4 on 1 and 32 DF, p-value: 5.417e-13</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Simple Linear regression      39

(Refer Slide Time: 10:04)

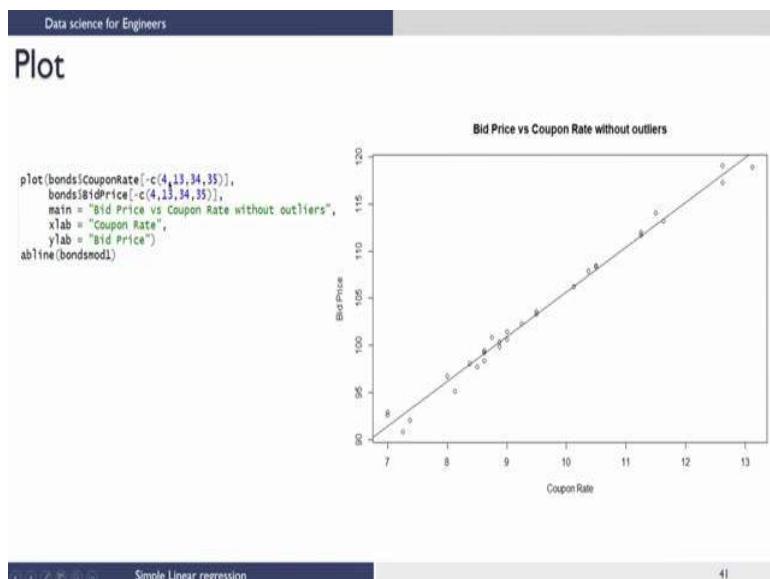


Now, I am removing the remaining points one by one. So, earlier I started by removing 13th point. Now, I am going to remove the 35th. So, let us see, what the R square value is, after removing the 35th point. So, the R square changes from 0.80 to 0.88. So, there is a quite big leap here as well.

So, after removing the 35th point, I am able to see a pretty good change in the R squared value. Now, let us look at what the R squared value is if I remove the fourth point. So, the R squared value improves from 0.88 to 0.98. So, that is also pretty good jump. So, now, these indices are for the old data. So, I also have one more index to remove, which is index 34. Now, let us see what happens, if we remove this.

So, after I remove the 34th point my R squared slightly increases; that is also from the 3rd decimal place. So, from 0.9852 it increases to 0.9891. So, the difference is not huge. We need not treat this point as an outlier by itself, because it does not improvise the model any further. So, now, after removing all these four points, we are going to plot the new regression line over the data.

(Refer Slide Time: 11:27)



So, on the left you can see, that I have removed the 4 index basically, 4 13 34 and 35. These points I have removed from my data and similarly, for bid price also, I have removed these points and I am going to fit the new model. So, bondsmod one does not have any outliers

now and I am going to plot the regression line over the data. So, our regression line fits the data pretty well, though there are some points, which are really away, but it does not change the nature of the slope drastically. So, this is a pretty good model and we have removed all the possible outliers that we thought were influencing the regression line.

(Refer Slide Time: 12:19)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. The main content area has a white background. At the top left, the word 'Summary' is displayed in a large, bold, black font. Below it, a bulleted list of six items is shown, each preceded by a small teal dot:

- Steps in building simple linear regression models
- Model summary
- Residual analysis
- Checking need for refinement
- Refined model building

At the bottom of the slide, there is a dark blue footer bar with several small white icons on the left (including arrows for navigation and a magnifying glass for search). In the center of the footer, the text 'Simple Linear regression' is written. On the far right of the footer, the number '42' is displayed.

So, to summarize in this three lectures, we looked at the steps, which are taken in building a simple linear regression model, we saw how to interpret the results from the summary, for these models. We looked at residual analysis. So, we looked at answering some of the question as how to treat outliers, we also saw how to identify significant coefficients in our model and how good our model is. We also saw the need for checking for refinement of existing models and then we built a refined model without any outliers.

Thank you.

**Data Science for Engineers**  
**Prof. Shankar Narasimhan**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 38**  
**Multiple Linear Regression**

Welcome everyone to this lecture on Multiple Linear Regression. In the preceding lectures we saw how to regress a single independent variable to a dependent variable. Particularly we were developing a linear model between the independent and dependent variable.

We also saw various measures by which we can assess the model that we built. In this lecture we will extend all of these ideas to multiple linear regression which consists of one dependent variable, but several independent variables. So, as I said that we have a dependent variable which we denote by  $y$  and several independent variables which we denote by the symbols  $x_j$ , where  $j = 1$  to  $p$ . There are  $p$  independent variables which we believe affect the dependent variable.

(Refer Slide Time: 00:49)

The slide has a dark blue header bar with the GyanData Private Limited logo. The main title 'Multiple Linear Regression' is centered above a list of bullet points. Below the list is a mathematical equation for the general linear model. At the bottom, there is a footer bar with navigation icons and the text 'Data Analytics'.

- ❑ Dependent variable ( $y$ ) depends on  $p$  independent variables  $x_j, j = 1, 2, \dots, p$
- ❑ General linear model

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \epsilon$$

- ❑ For  $i$ th observation

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \dots + \beta_p x_{p,i} + \epsilon_i$$

- ❑ Objective: Using  $n$  observations, estimate regression coefficients

We will try to develop a linear model between the dependent variable  $y$  and these independent  $p$  independent variables  $x_j$ ,  $j = 1$  to  $p$ . In general we can write this linear model as before we can say  $y$  the dependent variable =  $\beta_0$ , an intercept, +  $\beta_1$  times  $x_1$  +  $\beta_2$  times  $x_2$  and so

on up to  $b \beta$  times  $p \times p$ , where  $\beta_1, \beta_2, \beta_p$  represents the slope parameters or the effect of the individual independent variables on the dependent variable.

In addition we also have an error. This error is due to error in the dependent variable measurement of the dependent variable. In ordinary least squares we always assume that the independent variable measurements are perfectly measured and do not have any error whereas, the dependent variable may contain some error and that error is indicated as  $\epsilon$ . We do not know what this quantity is, we assume that it is a random quantity with 0 mean and some variance.

If we take the  $i$ th sample corresponding to this measurement of  $x_1$  to  $x_p$  and  $y$  corresponding  $y$  we can say that the  $i$ th sample dependent variable  $y_i = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon_i$ , the  $i$ th sample value of the independent variable 1. Similarly,  $x_2$  is  $\beta_2$  times  $x_2$  and so on + an error  $\epsilon_i$  that corrupts the measurement of  $y_i$  and so on for  $i = 1$  to  $n$ .

We assume we have small  $n$  number of samples that we have obtained. And our aim is to find the values, best estimates, of  $\beta_0, \beta_1, \beta_2$  up to  $\beta_p$  using these  $n$  sample measurements of  $x$ 's corresponding  $y$ . This is what we call multiple linear regression because we are fitting a linear model and there are many independent variables and we therefore, call the multiple linear regression problem.

(Refer Slide Time: 03:15)

The slide has a header 'GyanData Private Limited' and a title 'Multiple Linear Regression'. The content includes:

- ❑ Approach similar to simple regression  
*Minimize the sum of squares of the errors*
- ❑ Vector and matrix notations

$$\mathbf{y} = \begin{bmatrix} y_1 - \bar{y} \\ y_2 - \bar{y} \\ \vdots \\ y_n - \bar{y} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} x_{1,1} - \bar{x}_1 & x_{2,1} - \bar{x}_2 & \cdots & x_{p,1} - \bar{x}_p \\ x_{1,2} - \bar{x}_1 & x_{2,2} - \bar{x}_2 & \cdots & x_{p,2} - \bar{x}_p \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,n} - \bar{x}_1 & x_{2,n} - \bar{x}_2 & \cdots & x_{p,n} - \bar{x}_p \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

- ❑ The linear model in matrix form  
 $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, E(\boldsymbol{\epsilon}) = \mathbf{0}, Var(\boldsymbol{\epsilon}) = \sigma^2 \mathbf{I}$
- ❑ SSE  
 $S(\boldsymbol{\beta}) = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$

Again in order to find the best estimates of the parameters  $\beta_0$  to

$\beta_p$  we actually set up the minimization of the sum squared of errors. In order to set it up in a compact manner using vectors and matrices, we define the following notations. Let us define the vector  $y$  where which consists of all the  $n$  measurements of the dependent variable  $y_1$  to  $y_n$ , we have also done one further things we have subtracted the mean value of all these measurements from each of the observations.

So, the first one represents the first sample value of the dependent variable  $y_1$  - the mean value of  $y$  over all the measurements,  $\bar{y}$ . So, the first sample is mean shifted value of the first observation, the second coefficient or second value in this vector is the second sample value - the mean value of the dependent variable and so on for all the  $n$  observations we have.

So, these are the mean shifted values of all the  $n$  samples for the dependent variable. Similarly we will construct a matrix  $x$  where the first column corresponds to variable, independent variable 1. Again what we do is take the sample value of the first independent variable and subtract the mean value of the first independent variable. That means, we take the mean of all these  $n$  samples for the first variable and subtract it from each of the observations of the first independent variable.

So, the first coefficient here will be  $x_1$  represents the sample value of the first independent variable, first sample first independent variable, - the mean value of the first independent variable. And we do this for all  $n$  measurements of the first independent variable. Similarly we do this for the second independent variable and arrange it in the second column. So, this one represents the observation the first observation of the second independent variable - the mean value of the second independent variable and we do this for all  $p$  variables independent variables.

So, this particular matrix  $x$  that we get will be a  $n$  cross  $p$  matrix,  $n$  is the number of rows  $p$  is the number of columns. You can view the first row as actually the sample, first sample, of all independent variables for the first sample. Of course, we have being shifted that value. And the second row is the second sample and so on and each column represents a variable. So, first column represents the first independent variable and the last column represents the  $p$ th independent variable.

So, similarly we will represent all the coefficients  $\beta$  except  $\beta_0$  in a vector form  $\beta_1$  to  $\beta_p$  as a column vector. Here basically as a I am sorry a row vector. So,  $\beta_1$  is the first coefficient,  $\beta_p$  is the coefficient corresponding  $p$ th variable. So, we have  $\beta$  vector which is a  $p$  cross 1 vector we can also define  $\varepsilon$ , the noise vector, as  $\varepsilon_1$  to  $\varepsilon_n$  corresponding to all the  $n$  observations. Now having defined this notation we can write our linear model in the form  $y = x \beta + \varepsilon$ .

Notice that we have not included  $\beta_0$ . We have eliminated that indirectly by doing this mean subtraction I will show you how that happens. But you can take it that right now we have only interested in the slope parameters this linear model only involves the slope parameters  $\beta_1$  to  $\beta_p$ , does not involve the  $\beta_0$  parameter because that has been effectively removed from the linear model using this mean subtraction idea.

So, we can write our linear model compactly as  $y = x\beta + \varepsilon$  and we also make the usual assumptions about the error that it is a 0 mean vector in this case because it is a multivariate vector 0 is a vector. So,  $\varepsilon$  expected value  $\varepsilon = 0$  implies  $\varepsilon$  is a random vector with 0 mean and the variance, covariance matrix of  $\varepsilon$  is assumed to be  $\sigma^2$  identity.

$\sigma^2$  identity in this form it means all the  $\varepsilon$ s,  $\varepsilon_1$  to  $\varepsilon_n$ , have all have the same variance  $\sigma^2$  homoscedastic assumption. And we also assume that  $\varepsilon_1$  and  $\varepsilon_2$  are uncorrelated or  $\varepsilon_i$  and  $\varepsilon_j$  are uncorrelated if  $i$  is not equal to  $j$ , in which case we can write the covariance matrix of  $\varepsilon$  as  $\sigma^2 I$ .

Now, under this assumption we can go ahead and say we want to find the estimateds of  $\beta$  so as to minimize the sum square of the errors. So,  $\varepsilon^T \varepsilon$  is a compact way of saying the sum of all errors, error squared of all the errors, in all the  $n$  measurements. So, expanding this is nothing, but  $\sum \varepsilon_i^2 = 1$  to  $n$ , that is compactly written like this and this is what we want to minimize, but  $\varepsilon$  itself can be written as  $y - x\beta$ . So, we can write this whole thing as  $y - x\beta^T y - x\beta$ .

We want to minimize this which is a function of  $\beta$  by finding the best value of  $\beta$ . So, if we setup this optimization problem to minimize the sum squared errors to find  $\beta$  we will we can show. We can by differentiating that objective function with respect to  $\beta$  and setting it = 0 we get what are called the first order conditions and these first order conditions will result in the following set of linear equations. We will get  $X^T X$  into  $\beta = X^T y$ .

(Refer Slide Time: 08:54)

GyanData Private Limited

## Multiple Linear Regression

- Minimization of the SSE leads to the normal equations
- $$(\mathbf{X}^T \mathbf{X}) \hat{\boldsymbol{\beta}} = \mathbf{X}^T \mathbf{y}$$
- Assumption:  $(\mathbf{X}^T \mathbf{X})$  is of full rank  $p$  (invertible)
- The coefficients vector
- $$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}; \beta_0 = \bar{y} - \bar{x}^T \hat{\boldsymbol{\beta}}$$
- The properties of the estimators
  - $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$
  - $Var(\hat{\boldsymbol{\beta}}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$
- $\hat{\boldsymbol{\beta}}$  is the best linear unbiased estimator (BLUE)

Navigation icons: back, forward, search, etc.

Data Analytics

55

Now, this is a  $p$  cross, remember  $X$  is a  $n$  cross  $p$  matrix. So,  $X^T$  is  $p$  cross  $n$ . So, this is square matrix  $X^T X$  is a square matrix of size  $p$  cross  $p$  and multiplied by the  $p$  cross  $n$  vector and similarly it is  $p$  cross 1 on the right hand side. So, these are  $p$  equations in  $p$  variables. The linear equations in  $\beta$  x is all known y is known. So, right hand side is like the if you, are what we, reinterpret this as a times some  $x = b$ . It is a set of linear equations  $p$  equations in  $p$  variables which can be easily solved if  $a$  is invertible.

So, we assume that  $X^T X$  is a full rank matrix invertible. The meaning of this will become little clearer later, and if it is not invertible we will have to do other things which we will again talk in another lecture. But at for the time being let us assume that  $X$  transpose  $X$  which is a square matrix is invertible it is a full rank matrix and then you can easily find the solution for  $\beta$  solve this linear set of equations by taking  $a^{-1}b$  which is exactly  $X^T X$  inverse  $X^T y$ . So,  $\beta$  the coefficient vector can be found by this thing.

And this is the solution that minimizes the sum squared errors, this objective function that we have written. So, once we get  $\beta_1$  the slope parameters  $\beta_0$  can be estimated as the mean value of  $y$  - the mean vector  $X^T$  times the slope parameter. Notice that this is very similar to what we have in the univariate case where it says  $\beta_0$  estimate is nothing but  $\bar{y} - \bar{x}$  into  $\beta_1$ . So, it is very similar to that you can see.

You can also compare the solution for the slope parameters, for the, with the univariate case which says  $\beta_1$  is  $SXY$  divided by  $SXX$ . Notice that  $X$  transpose  $y$  represents  $SXY$  and  $X^T X$  represents  $SXX$  in the univariate case you were diving  $SXY$  by  $SXX$  in the multivariate case

division is represented by an inverse. So, you get  $X^T X$  inverse terms times  $X^T y$ . So, you can see the it is very very similar to the solution for the univariate case except that these are matrices and vectors and therefore, you have to be careful. You cannot simply divide it as matrix times inverse times a vector that is a solution for  $\beta$  which is slope parameters.

You can also estimate  $\beta_0$  and  $\beta_1$  by doing what is called augmentation of the  $X$  vector with a constant value 1 1 1 in the final thing, but I did not use that approach because the mean subtraction approach is a much better approach for estimating whether if for estimating  $\beta_0$  and  $\beta$ ,  $\beta$  slope parameters because this is applicable even to another case called the total least squares.

The augmentation approach is valid only for ordinary least squares you cannot use it for total least squares which we will see again later. So, that is why I use the mean subtraction route in order to obtain the estimates of the slope parameter first followed by the estimation of  $\beta_0$  using the estimates of the slope parameters in this manner.

Now, you can also derive properties of these parameters  $\beta$ . We can show that the  $X$  vector value of  $\hat{\beta}$  is  $\beta$  which just means it is an unbiased estimate just as in the univariate case and we can also get the variance of this  $\hat{\beta}$  the in this case it is a covariance matrix because it is a vector and we can show that the covariance matrix is  $\sigma^2$  times this  $X$  transpose  $X$  inverse.

Now, again you can go back and look at the univariate case. There the variance of  $\beta_1$  slope parameter will be  $\sigma^2$  by  $S_{XX}$  in this case it is  $\sigma^2$  into  $X^T X$  inverse. So,  $X^T X$  represents  $S_{XX}$ .  $\sigma^2$  is the variance of the error corrupting the dependent variables. We may have a priori (Refer Time: 13:35) knowledge sometimes in most cases we may not be able to know this value of  $\sigma^2$  we may not be given this. So, we have to estimate the  $\sigma^2$  from data and we will show how to get this.

These two parameters that actual we can show the first parameter says that the estimates of  $\beta_1$  the slope parameters are unbiased. So,  $\hat{\beta}$  are unbiased estimator it is an unbiased estimator of the of the true value  $\beta$ . Moreover you can show that among all linear estimators because  $\hat{\beta}$  is a linear function of  $y$ . Notice that  $(X^T X)^{-1} X^T$  is nothing but matrix which basically multiplies the measurements  $y$ . So,  $\hat{\beta}$  can be interpreted as a linear combination of the measurements. Therefore, it is known as a linear estimator.

Among all such linear estimators we can show that  $\hat{\beta}$  has the least variance. Therefore, it is called a blue estimator or a unbiased estimator with the best linear unbiased estimator that is what it blue represents, best in the sense of having the least variance.

(Refer Slide Time: 14:52)

□ Estimate of the error variance

$$\hat{\sigma}^2 = \frac{\sum(y_i - \hat{y}_i)^2}{n-p-1}$$

where  $(n-p-1)$  is the degrees of freedom (df)

□  $1-\alpha$  confidence intervals for  $\beta_j, j = 0, 1, \dots, p$

$$\beta_j \in [\hat{\beta}_j - t_{(n-p-1, \alpha/2)} s.e.(\hat{\beta}_j), \hat{\beta}_j + t_{(n-p-1, \alpha/2)} s.e.(\hat{\beta}_j)]$$

$t_{(n-p-1, \alpha/2)}$  is the  $(1 - \alpha/2)$  percentile point of the  $t$ -distribution with  $(n-p-1)$  df

$$s.e.(\hat{\beta}_j) = \hat{\sigma} \sqrt{c_{jj}}$$
$$C = (X^T X)^{-1}$$

56

Now, we can also estimate as I said  $\sigma^2$  from the data and that  $\sigma^2$  estimate is nothing but the after you fit the linear model you can take the predicted value for the  $i$ th sample from the linear model and compute this residual  $y_i - \hat{y}_i$  which is the measured value - the predicted value for the  $i$ th sample, square it take the sum of all possible samples,  $n$  samples, divided by  $n - p - 1$ . Again if you go back to your linear case univariate case you will find that the denominator is  $n - 2$ . Here you have  $n - p - 1$  because you are fitting  $p + 1$  parameters  $p$  is slope parameters + 1 o set parameter.

Therefore out of the  $n$  measurements  $p + 1$  are taken away for the deriving the estimates. Only the remaining things are the degrees of freedom or the variability in the residuals is cost by the remaining  $n - p - 1$  measurements and that is why you are diving by  $n - p - 1$  whereas, in the univariate case you would have divided by  $n - 2$  because you are estimating only two parameters there.

So, you can see a one to one similarity between the univariate regression problem and the multi multiple linear regression problem in every derivation that we have given here. Now, once we have estimated  $\hat{\sigma}$ , the variance of the error used from the data you can go back and construct confidence intervals for each slope parameter we can show that the true slope parameter lies in this confidence interval for any confidence interval you may choose  $1 - \alpha$ ,  $\alpha$  represents like a level of significance.

So, if you say  $\alpha = 0.05$ ,  $1 - \alpha$  would represent 0.95. So, that will be a 95 percent confidence interval. Correspondingly I will find the critical value from the  $t$  distribution  $n$  with  $n - p - 1$  degrees of freedom

and this represents  $\alpha$  by 2 the upper lower value probability value from the t distribution and this is the upper critical value where the probability area under the curve beyond the value is  $\alpha$  by 2. So,  $n - p - 1$  represents the degrees of freedom notice that in the univariate case it would have been  $n - 2$ , very very similar.

So, the confidence interval for  $\beta_j$  for any given  $\alpha$  can be computed using this particular formula and the term here se of  $\beta_j$  represents the standard deviation of the estimate of  $\beta_j$  and that is given by the diagonal element, diagonal element here of this quantity with  $\sigma^2$  replaced by the estimate here.

So, we have computed the standard deviation of the, of the parameter  $\beta_j$  estimated parameter  $\hat{\beta}_j$  by using the estimated value of  $\sigma$  multiplied by the diagonal element of  $X^T X$ . So, we are fitting the diagonal elements of the covariance matrix of  $\beta$  parameters that is all we have done. So, this represents the diagonal element or the square root of the diagonal element which represents standard deviation of the estimated value of  $\beta$  which is what is used in order to construct this confidence interval.

So, every one of this can be computed from the data as you can see and you can construct. Now, the confidence level can later be used for testing whether the estimated parameter  $\beta$  is significant or insignificant as we will see later.

(Refer Slide Time: 18:39)

❑ Multiple correlation coefficient

$$Cor(y, \hat{y}) = \frac{\sum(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum(y_i - \bar{y})^2} \sqrt{\sum(\hat{y}_i - \bar{\hat{y}})^2}}$$

❑ The coefficient of determination  $R^2$

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

❑ Adjusted R-squared,  $R_a^2$

$$R_a^2 = 1 - \frac{SSE/(n-p-1)}{SST/(n-1)}$$



Now, we will, can also compute the correlation between  $y$  and  $\hat{y}$  which tells you whether the predicted value from the linear model is, resembles or closely related to, the measured value. So, typically we will draw a line between the  $y$  the measured value and the predicted value and see whether it is these things fall on the 45 degree line and if it does then we think that the fit is good. Another way of doing this is to find the correlation coefficient between  $y$  and  $\hat{y}$  which is simply using

the standard thing  $y_i - \bar{y}$  multiplied by  $\hat{y}_i - \bar{y}$  hat bar. Summed over all quantities divided by the standard deviation of in  $y$  and the standard deviation in  $\hat{y}$  that is for normalization.

We could also use the coefficient of determination, R squared, just as we did for the univariate case. We can compute R squared as  $1 - \frac{\text{sum squared error}}{\text{sum squared total}}$ . Which is nothing, but numerator is  $y_i - \hat{y}_i$ , the residual squared divided by  $y_i - \bar{y}$  squared which is the variance in  $y$  basically. So, if we take  $1 - \frac{\text{sum squared error}}{\text{sum squared total}}$  this we will, actually we can show whether using the independent variables have we been able to get a better t. If we have obtained a very good t then the numerator will be close to 0 and R squared will be close to 1.

On the other hand if we are not improved the t because of x's any of the x's, then the numerator will be almost equal to the denominator and therefore, this one will be close to 0. So, value of R square close to 1 as before represents indication of a good linear t whereas, a value close to 0 indicates the t is not good. We can also compute adjusted R square to account for the degrees of freedom notice that the numerator has  $n - p - 1$  degrees of freedom whereas, the denominator has  $n - 1$  degrees of freedom therefore, we can

do an adjusted R squared which divides the SSE by the appropriate degrees of freedom.

We can say this is the error due to per degree of freedom that is there in the t whereas, the denominator represents the error because we have fitted only the o set parameter there are  $n - 1$  degrees of freedom this is the error per degree of freedom. So, this kind of a thing is also a good indicator instead of using R squared we can use adjusted value of R square. So, these are all very very similar again to the univariate linear regression problem.

(Refer Slide Time: 21:17)

GyanData Private Limited

## Multiple Linear Regression

- Fitted model is adequate or can be reduced further?
  - Test significance of individual coefficient  $\hat{\beta}$
  - A general unified test on the full model (FM) vs the reduced model (RM)
- Hypothesis testing
  - $H_0$ : Reduced model is adequate
  - $H_1$ : Full model is adequate



So, we can use, we can check R squared and see whether the values close to one and if it is we can say maybe linear model is good to fit the data, but that is not a confirmatory test. We have to do the residual plot as we did in linear regression, univariate linear regression, and that is what we are going to do further. So, we are going to find whether the fitted model is adequate or it can be reduced further. What this reduced further means we will explain. In the univariate case there is only one independent variable, but here there are several independent variables. Maybe not all independent variables have an effect on y. Some of the independent variables may be irrelevant. So, one way of trying to find whether a particular independent variable has an effect is to test the corresponding coefficient.

Notice we have already defined the confidence interval for each coefficient and we can see whether the confidence interval contains 0, in which case we can say the corresponding independent variable does not have a significant effect on the dependent variable and we can perhaps drop it. Or, we can also do what we call the test, F test, just as we did univariate regression problem we test whether the full model is better than the reduced model.

The reduced model contains no independent variables whereas, the full model can contain all or some of the independent variables. You can do many kinds of test and we will do this. So, we can test whether the reduced model which contains only the constant intercept parameter is a good fit as opposed to including all the independent variables, some or all the independent variables, that is what we call the full model.

(Refer Slide Time: 23:02)

**GyanData Private Limited**

## Multiple Linear Regression

- ❑ Testing two models: RM with  $k$  parameters
- ❑ F-statistic

$$F_o = \frac{[SSE(RM) - SSE(FM)] / (p+1-k)}{SSE(FM) / (n-p-1)}$$

Degrees of freedom

- ❑ Note that  $SSE(RM) \geq SSE(FM)$
- ❑ For  $\alpha$ -significance level: Reject  $H_0$  if  
 $F_o \geq F_{(p+1-k, n-p-1; \alpha)}$   
 where F-statistic for the given dfs from the table

We will consider a specific case here where we do the F test statistic for the case when we have a reduced model and compare it with the full model. The reduced model we will consider with  $k$  parameters specifically let us consider the reduced model with only one parameter which means that we have only the constant intercept parameter we will not include any of the independent variables. And compare it with the full model which contains all of the independent variables including the intercept.

So, the reduced model is one which contains only the  $\alpha$  set parameter and no independent variables the full model is a case where we consider all the independent variables and the intercept parameter. So, the number of parameters we are estimating in the reduced model is only 1, so  $k = 1$  and the full model is the case where we have all the independent variables  $p$  independent variables. So, we are estimating  $p + 1$  parameters in the full model.

So, what we do is perform a fit and compute the sum squared errors which is nothing but the difference between  $y$  the measured value and the predicted value. So, we will first take the model containing only the  $\alpha$  set or the intercept parameter and estimate. In this case of course,  $y$  bar will be the best estimate. And we will compute sum squared errors which is nothing but the variance of the measured measurements for the dependent variable. Then we will also perform a linear regression containing all the parameters, independent variables, and in this case we will if we compute the difference between  $y$  and  $y$  predicted and take the sum squared errors that is the SSE of the full model.

So, when we want to compare whether we want to accept the full model as compared to the reduced model what we do is take the difference in the sum squared errors remember the sum squared errors for the reduced model will become greater than the sum squared errors for the full model because the full model contains more number of parameters and therefore, you get a better fit.

So, the difference in the fit which is difference in the sum squared errors between the reduced model fit and the full model fit that is the numerator, divided by what we call the degrees of freedom. Notice the full model as  $p + 1$  parameters  $p$  independent variable + the  $\alpha$  set and the reduced model in this particular case contains only 1 parameter, so,  $k = 1$

So, the degrees of freedom will be  $p$ . So, you divide this difference in the sum squared errors by  $p$  denominator is the sum squared errors of the full model which contains  $n - p - 1$  degrees of freedom because  $p + 1$  parameters have been fitted therefore, the degrees of freedom is the total number of measurements -  $p - 1$ . So, we divide the sum squared

errors for the denominator by the number of degrees of freedom and then take this ratio as defined and that is your F statistic.

Now, in order to reject, if we want to reject the null hypothesis, or if we want to test the null hypothesis against this alternative we find the test criteria for the  $\alpha$  level of significance. We will take it from the F distribution where the numerator degrees of freedom is  $p + 1 - k$  for this particular case it is exactly  $p$  and the denominator degrees of freedom is  $n - p - 1$  and  $\alpha$  level of significance we use and we compute the test criteria, critical value from the F distribution.

Then we compare the test statistic with the critical value and if the test statistic exceeds the critical value at this level of significance ,then we reject the null hypothesis. That is we will say the full model is better choice and the independent variables do make a difference. And this is a standard thing that R function will provide. This particular comparison between the reduced model which has no independent variables and the full model which contains all the independent variables in multi linear regression.

Of course, you can choose different reduced models and compare with the full model. For example, you can take the reduced model by leaving out only one of the independent variables. So, that will have  $p$  parameters, we can compare it with the full model and again perform a test to decide whether the inclusion of that independent variable makes a difference or not.

So, this kind of combination can be done depending on what stage you are and that will be using in what we call the sequential method for subset selection that will be discussed in the later lecture. But essentially the R functions only provide a comparison between the reduced model which contains no independent variable and the full model which contains all of the independent variables. Let us go through simple example in order to what you call revisit these ideas.

(Refer Slide Time: 27:51)

Menu pricing in Restaurants of NYC

$y$  : Price of dinner  
 $x_1$ : Customer rating of the food (Food)  
 $x_2$ : Customer rating of the décor (Décor)  
 $x_3$ : Customer rating of the service (Service)  
 $x_4$ : If the restaurant is east or west (East)

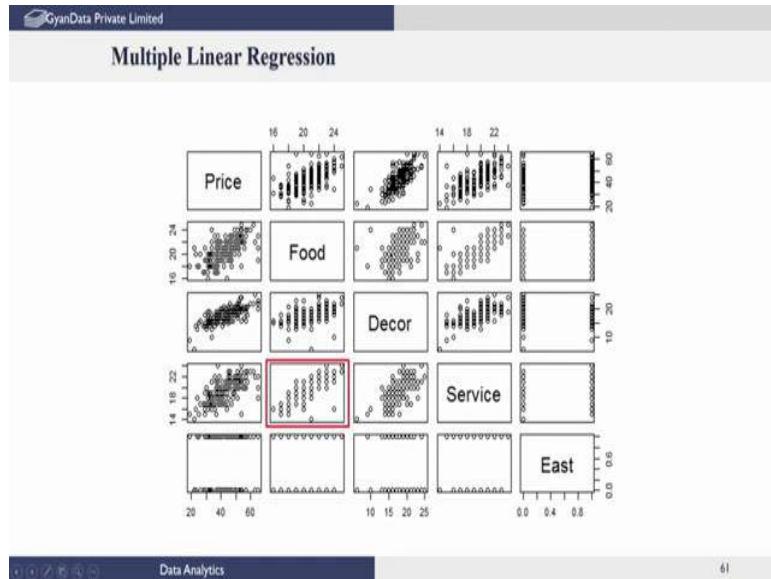
Objective: Build a model

So, in this case we have what is called the price data where customers are being asked to rate the food and the other aesthetics of a particular restaurant and we also and cost of the particular dinner also data(Refer Time: 28:15) obtained for these restaurants. And the location of these restaurants whether on they are on the east side of a particular street in New York or the west side. Typically New York Westside is probably a little poorer whereas, the east side probably is a little richer neighborhood.

So, location of the restaurant also would indicate, would have a effect on, the price. So, these are the four independent variables people data was obtained on. The quality of the food, the decor and service all this was rated by the customers and the location of this restaurant and the price of dinner in that served in that restaurant was also taken.

So, you would expect that the quality of the food the service level all of this would have a very direct influence on the price in the restaurant and a linear model was built between  $y$  and the independent variables  $x_1$  to  $x_4$ .

(Refer Slide Time: 29:17)



So, before we build a model we do a scatter plot as usual and visualization. And here you because there are several independent variables we have not just one plot scatter plot between  $y$  and  $x_1$ . For example, in this case remember price is  $y$ ,  $y$  versus  $x_1$  this particular plot shows the correlation or the scatter plot for  $y$  versus  $x_1$  or  $y$  versus food, price versus food. The second one is the scatter plot will be price and decor.

The third one is the scatter plot between price and service and the last one is price versus location. And similarly you can actually develop a scatter plot between food and decor which is here or food and service and so on.

Even though we consider all these variables that we have obtained like food ,decor, service, location as independent. It is possible when we select these variables they are not truly independent there might be interdependencies between

the what we call so called independent variables that can give rise to problem in regression which we will see later the what we call the effect of collinearity. But a scatter plot may reveal some dependencies, inter dependencies, between the independent so called independent variables.

So, for example, if we look at the scatter plot between food and decor it is seems to be completely randomly distributed this does not seem to be any quite correlation. However, food and service seems to be very strongly correlated there seems to be a linear relationship between food and service.

So, perhaps you do not need to include both these variables we will see later that that it is true. But in this just a scatter plot itself reveals

some interesting features and so we will now go ahead and say perhaps a linear mode between price and food and decor is, seems to be pointed out or, indicated by this scatter plots let us go ahead and build one.

(Refer Slide Time: 31:17)

```

GyanData Private Limited
Multiple Linear Regression

Regression output from R

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -24.023800 4.708359 -5.102 9.24e-07 ***
Food 1.538120 0.368951 4.169 4.96e-05 ***
Decor 1.910087 0.217005 8.802 1.87e-15 ***
Service -0.002727 0.396232 -0.007 0.9945
East 2.068050 0.946739 2.184 0.0304 *

Signif. codes: '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.738 on 163 degrees of freedom
Multiple R-squared: 0.6279, Adjusted R-squared: 0.6187
F-statistic: 68.76 on 4 and 163 DF, p-value: < 2.2e-16

 $\hat{y}_i = -24.024 + 1.538x_1 + 1.910x_2 - 0.003x_3 + 2.068x_4$

```

And if we apply the R function lm to this data set and we examine the output. We will get this output from R and it tells that the intercept term is - 24.02 and the slope parameters, the coefficient multiplying food is 1.5, the coefficient multiplying decor is 1.9 and so on so forth.

It also gives you the standard error for each coefficient as well as the offset parameter which is nothing but the  $\sigma$  value for the estimated quantities and it also gives you the probability values p values as we call them. And notice if the p value is very low it means that this coefficient is significant. We cannot take it that this value is. Any low value of this indicates that the corresponding coefficient is significantly different from 0.

So, in this case the first three has very low p values and therefore, they are significant, but service has a high p value therefore, it seems to indicate that this coefficient is insignificant is equal almost = 0 that is what this indicates. If you look at the east which is this independent location parameter that as does not have a very low p value. But it is still not bad 0.03 and therefore, it is significant only is insignificant only if you take a level of significance of 0.025 or something like that. If you take 0.1 or 0.05 and so on you will still consider this east ,this coefficient, to be significant and that is what this is basically pointing out this star indicates that.

So, now we will go ahead and try to actually look at the F value also, the F statistics says that the full model as compared to the reduced model of using only the intercept is actually significant. Which means

the constant model is not good and including these variables results in a better t or explanation of the price and therefore, you should actually include this. Whether you should include all of them or only some of them we can do different kinds of test to find that.

What we have done in this particular case is only compare the model with-out any of these independent variables which is called the constant model with all of these variables included that is the only two model comparisons we have made. The reduced model is one containing only the interceptor and the full model is one which contains intercept and all four independent variables and thus the p value it has given the corresponding F statistic.

So, we are saying that including these independent variables is important in explaining the price. But it may turn out that all of them is not necessary and that we will we will examine further. So, the corresponding t that we obtained is this. As I said that from the, what you call, the confidence interval for the slope parameter for service, we can say that we can remove this it is insignificant and perhaps we can remove this and try the t. For the time being let us actually remove this and try the t.

(Refer Slide Time: 34:42)

|             | Estimate | Std. Error | t value | Pr(> t )     |
|-------------|----------|------------|---------|--------------|
| (Intercept) | -24.0269 | 4.6727     | -5.142  | 7.67e-07 *** |
| Food        | 1.5363   | 0.2632     | 5.838   | 2.76e-08 *** |
| Decor       | 1.9094   | 0.1900     | 10.049  | < 2e-16 ***  |
| East        | 2.0670   | 0.9318     | 2.218   | 0.0279 *     |

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.72 on 164 degrees of freedom  
 Multiple R-squared: 0.6279, Adjusted R-squared: 0.6211  
 F-statistic: 92.24 on 3 and 164 DF, p-value: < 2.2e-16

$$\hat{y}_i = -24.027 + 1.536x_1 + 1.910x_2 + 2.067x_4$$

*Caution:* Removing several predictors may have a dramatic effect on the coefficients in the reduced model

We have done that. We have only included now food, decor and east and done the regression again and it turns out that regression thing is still what you call the R squared value is improved not improved significantly, but not reduced and F value is significant and we get the more or less the same coefficients for the other parameters also the intercept and the slope parameters.

It indicates that  $x_3$  is not adding any value to the prediction of y. The reason for this as we said if you look at the scatter plot service and

food are very strongly correlated therefore, only either food or service needs to be included in order to explain price and not both right. And in this case service is being removed, but you can try removing food as the variable and try to fit between price, decor and decor service and east and you will find that the regression is as good as retaining food and eliminating service.

(Refer Slide Time: 35:45)

The screenshot shows a presentation slide with the following content:

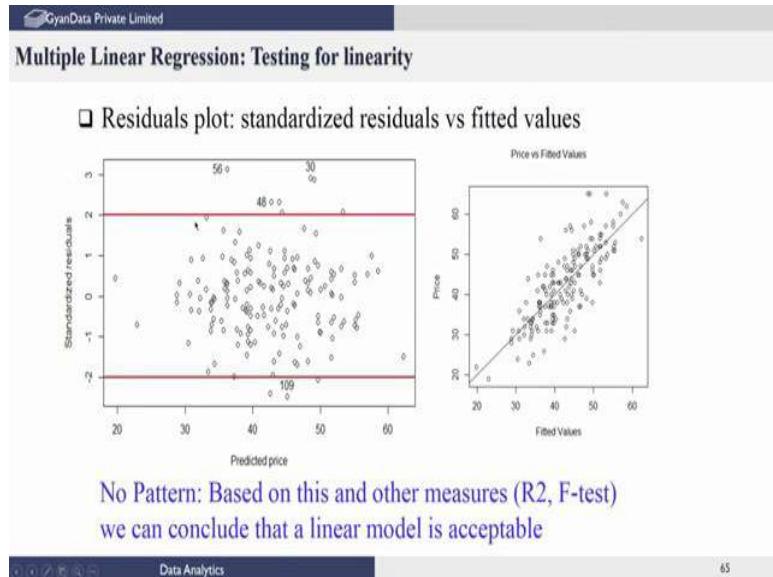
- Residual plots: Standardized residuals for assessing
  - Linear vs nonlinear model
  - Normality of the errors
  - Homoscedastic vs heteroscedastic errors

A green curly brace on the right side groups the last three items under the heading "Similar to Simple regression".

At the bottom right of the slide, there is a video player showing a man speaking against a green background. The video player has standard controls (play/pause, volume, etc.) and the text "Data Analytics" below it.

R squared value and the f statistics seems to indicate that we can go ahead with the linear model, but we should further examine the standardized residual plot for concluding whether the linear model is or not. There should no pattern in the residuals. So, let us actually do the residual plot.

(Refer Slide Time: 36:00)



Here we have taken the standardized residuals and plotted it against the, what is called the predicted price value or the fitted value. Remember this is  $\hat{y}_i$ ,  $\hat{y}_i$  has only one variable. So, you need to generate only one plot and we have also shown here the, in red lines, the confidence interval for the standardized residuals and anything above this outside of this interval indicates outliers. So, for example, 56, sample number 48, sample number 30 and 109 and so on so forth may be possible outliers and, but there is no pattern in the standardized residuals.

It is spread randomly within this boundary and therefore, we can say since there is no pattern a linear t is acceptable. So, here the quality of the t is shown here. So, the actual price measured value versus the  $\hat{y}_i$  the predicted value is shown and a linear model seems to explain the data reasonably well.

The last thing is we have these outliers if you want to improve the t you may want to remove let us say the outlier which is farthest away from the boundary.

For example, you may want to remove 56 and redo the linear regression multi multiple linear regression and again repeat it until there are no outliers. That will improve the R squared value and the fit quality of the fit little more.

So, we have not done that we leave this as exercise for you. So, what we have done is we have seen that whatever was valid for the univariate regression can be extended to the multi multiple linear regression except that scalars there will get replaced by vectors and

matrices corresponding. What was a variance there it will become a variance covariance matrix here, what was a vector there scalar there might mean scalar might become a mean vector here.

So, you will see a one to one correspondence, but the residuals plot and interpretation of confidence interval for  $\beta$  all of this, the F statistic more or less similar. Except that understand in the multiple linear regression there are several independent variables all of them may not be relevant.

We may be able to take only a subset and I will actually handle subset selection as a separate lecture. For the time being we are just done a significance test on the coefficient in order to identify the irrelevant independent variables, but there are better approaches and we will take it up in the following lectures.

**Data science for Engineers**  
**Prof. Shankar Narasimhan**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture – 39**  
**Cross Validation**

Welcome everybody to this last lecture on regression. In this lecture I am going to introduce concept called Cross Validation which is a very useful thing in model building.

(Refer Slide Time: 00:28)

The slide has a dark blue header bar with the text 'GyanData Private Limited'. The main title 'Motivation' is centered in a light gray box. Below the title is a bulleted list of points. Some text and equations are interspersed within the list. At the bottom of the slide, there is a dark blue footer bar with icons and the text 'Data Analytics' and '89'.

- How to select the optimal number of meta or hyper-parameters of a model?
  - Number of principal components in principal components analysis
  - Number of clusters in K-means clustering
  - Number of terms ‘ $n$ ’ in polynomial or nonlinear regression
$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots + \beta_n x^n$$
(equivalent to multilinear regression by treating  $x, x^2, \dots, x^n$  as different variables)
- MSE of training data set not useful as a measure
  - MSE will decrease with increasing number of parameters (can be reduced to zero)
- Use cross validation on a validation data set to determine optimal number of parameters

The main purpose of cross validation is to select what we call the number of meta parameters or hyper parameters of a model. Although we have not introduced principal component analysis in this series of lectures, nevertheless when you learn it in a higher advance course on data analytics, you will come across this idea of principal components and one of the problems in principal component analysis is to select the number of principal components that are relevant. We call this a hyper parameter or a meta parameter of the model.

Similarly, later on in this course you will come across clustering. In particular you will come across K means clustering and here again, you have to choose the number of clusters required to group the data and the number of clusters is called the meta parameter of the clustering

modeling. Similarly, if you are building a non-linear regression model. For example, let us take a polynomial regression model where the dependent variable  $y$  is written as a polynomial function of the independent variable  $x$ . Let us assume there is only one variable  $x$ . You can write the regression model, non-linear regression model, as  $y = \beta_0 + \beta_1$  which is the linear part. You can also include non-linear terms such as  $\beta_2 x^2$  and so on up to  $\beta_n x^n$ , where  $x^2 x^3$  and so on are higher powers of  $x$ .

This is known as a polynomial model. Notice this, the polynomial model, can also be the parameters of this model can be obtained using multi linear regression. If you take  $x$  as a variable  $x^2$  as a different variable and  $x_n$  as a different variable computed from data that we are given treat them as different variables, then you can use multi linear regression methods in order to estimate the parameters  $\beta_0 \beta_1$  and so on up to  $\beta_n$ . Here again, we have to decide how many powers of  $x$  we have to choose.

So, if you choose higher powers of  $x$ , then corresponding to each power you got a extra parameter that you need to estimate. For example, in this case you have  $n + 1$  parameters if you have chosen  $x$  power  $n$  as your highest degree of the polynomial. The choice of the degree of the polynomial to  $t$  is again the a meta parameter of the non-linear regression model. So, in all of these this cases, you have to find out the optimal number of meta parameter, optimal number of parameters of the model that you need to use in order to obtain the best model.

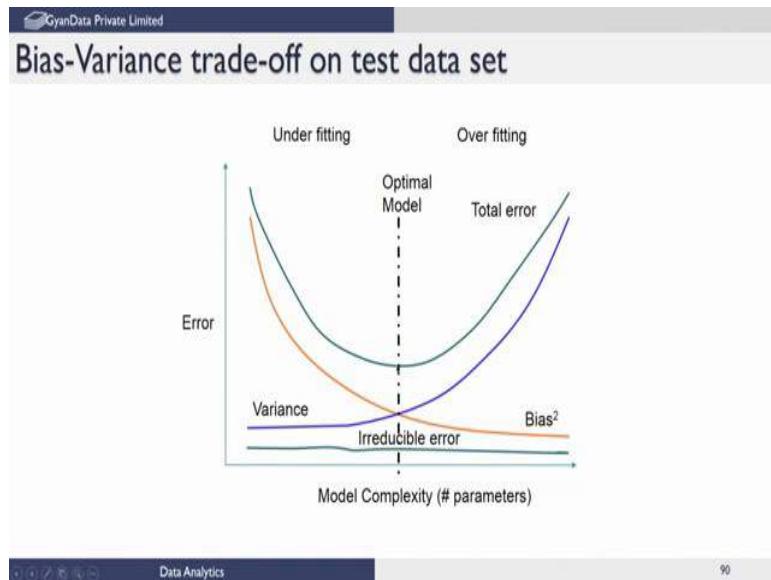
Now, you might think that it is easy to actually obtain this by looking at the mean squared error of the training data. So, for example, if you take this polynomial model fit it and compute the mean squared error in the training data. That is nothing but the difference between  $y$  and  $\hat{y}$  after you have found out best fit parameters of  $\beta_0, \beta_1$  up to  $\beta_n$ , you can predict for every data by substituting the value of  $x, x^2$  and so on and you predict  $\hat{y}$  and you compute the difference  $y - \hat{y}$  and sum over squared of all this is called the mean squared error.

So, the mean squared error of the training data you might think is useful as a measure for finding the optimal number of parameters, but that is not true because as we increase the number of parameters, if you keep adding higher order terms, you will find the mean squared error decreases. So, ultimately you can get the mean squared error to decrease to 0 as you increase the number of parameters this is called over-fitting.

So, you can always get the mean squared error of training data to 0 by choosing sufficient number of parameters in the model. Therefore, you cannot use the training data set in order to find out the best number of, optimal number of, parameters to use in the model. So, we do something called cross validation. This has to be done on a different

data set that is not used in the training. We call this the validation data set and we use the validation data set in order to decide the optimal number of parameters or meta parameters of this model. So, we will use this polynomial regression model as an example throughout in order to illustrate this idea of cross validation.

(Refer Slide Time: 04:43)



So, schematically what happens when you actually increase the model complexity or the number of meta parameters of the model. So, you will find that the mean squared error, On the test set continuously decreases. So, it will goes to a 0 as we said on the training set; however, on the validation set what will happen is, if you look at the mean squared error on the validation set, that will initially decrease as you increase the number of parameters, but beyond a certain point the mean squared error on the validation set will start increasing. So, the optimal number of parameters you should choose or the model complexity you should choose, corresponds to the minimum value of the mean squared error on the validation set and this is called the optimal model. If you choose less number of parameters than the optimal model, we call this under fitting. On the other hand, if you use more parameters in your model, than the optimal model value is called over fitting.

So, over fitting, basically means you are using unnecessarily more parameters than necessary to explain the data. On the other hand, if you use less parameters, you actually or not sufficiently your model is not going to be that accurate. Typically, there are two measures for determining the quality of the model. One is called the bias in your prediction error and if you know if you increase the number of

parameters of the model, this bias squared of the bias term will start decreasing.

However, the variability in your model predictions that will start increasing as you increase the model complexity or number of parameters. So, it is basically that trade off between these two that gives rise to this minimum value of the MSE on the validation set.

That is what you are looking for. So, want this optimal trade off between the bias which keeps reducing as you increase the number of parameters and the variance which keeps increasing as the number of parameters in the model increases. So, this is what we are going to find out by cross validation.

(Refer Slide Time: 06:47)

**GyanData Private Limited**

## Training and Validation data sets

- For large data sets divide data set into training data set (~ 70% of the samples) and remaining validation/test data
  - Training set:  $\{(\mathbf{x}_1, y_1); (\mathbf{x}_2, y_2); \dots; (\mathbf{x}_n, y_n)\}$
  - Test set:  $(\mathbf{x}_{0,i}, y_{0,i}) : i = 1 \dots n_t$ , observations
- Training error rate
 
$$MSE_{Training} = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \hat{\beta})^2 \quad \begin{matrix} \text{Not of our interest} \\ \text{for predictive} \\ \text{ability of the model} \end{matrix}$$
- Test error rates
 
$$MSE_{Test} = \frac{1}{n_t} \sum_{i=1}^{n_t} (y_{0,i} - \mathbf{x}_{0,i}^T \hat{\beta})^2 \quad \begin{matrix} \text{Of our interest} \end{matrix}$$

*Data scarcity: Test data are not available*

So, if you have a large data set, then you can always divide this data set into 2 parts; one used for training typically 70 percent of the data samples you will use for training and the remaining, you will set apart for the validation. So, let us call the samples that you use for training as  $x_1, y_1, x_2, y_2$  and so on where  $x$  represents the independent variable,  $y$  is the dependent variable. And the validation set, we will denote it by the symbols  $x_{0,i}$  and  $y_{0,i}$  where there are  $n_t$  observations in the validation set. So, typically as I said if you have a large number of samples, you can set apart 70 percent of the samples for training and the remaining 30 percent, we can use for validation. Now, you can always of course, define the mean squared error in the training set after building the model.

So, this is nothing but the difference between the measured, observed value of the dependent variable - the predicted value after you have estimated the parameters, let us say using least squares regression. So, this is the prediction error on the training data square over all the

samples and taken the average, that is the mean squared error that we have seen before.

You can do a similar thing for the validation set also. You can take the difference between the measured value or observed value in the validation sample set - the predicted value for the validation samples and again you can take the sum square difference between the observation - the predicted value for the validation set squared over all samples on the averaged average value.

So, that is called the mean squared error on the test or validation. So, this particular term as I said the MSE on training is not useful for the purpose of deciding on the optimal number of parameters of the model; however, this test MSE test or the mean squared error or the validation data set is the one that we are going use for finding the optimal number of parameters of the model.

(Refer Slide Time: 08:53)

The slide has a header 'GyanData Private Limited' and a title 'Validation Set Approach'. It contains a bulleted list of points and a diagram illustrating the division of data into training, validation, and test sets.

- Enough data: (1) Training set, (2) Validation set, and (3) Test set
- Not enough data: Generate validation sets from a training set
- Validation set approach: Divides (often randomly) the training set into two parts

Diagram illustrating the division of data:

|                                    |         |                |
|------------------------------------|---------|----------------|
| 1 2 3 4                            | n       |                |
| A training set                     | 1 2 3 4 | n <sub>t</sub> |
| A validation set (or hold-out set) | 1 2 3 4 | n <sub>v</sub> |

- Use training set, to fit the model
- Use validation set, to predict validation set errors

Provides an estimate of test error rates

Navigation icons and slide number '92' are visible at the bottom.

So, as I said, if you have large number of amount of samples, then you can actually divide it into a training set, a validation set for finding the optimal number of parameters of a model and finally, if you want to assess, how good your optimal model is you can run it on a test set. So, typically you take the data set and you divide it into three parts, one the training set the validation set where you are trying to use for finding the optimal number of parameters and finally, the test set for to see whether the optimal model you have built is good enough.

We will not worry about the test set in this particular lecture. We will only worry about this validation set. Unfortunately, if you do not have large number of samples, so you would actually generate a validation set from the training set itself and we will see how to do this

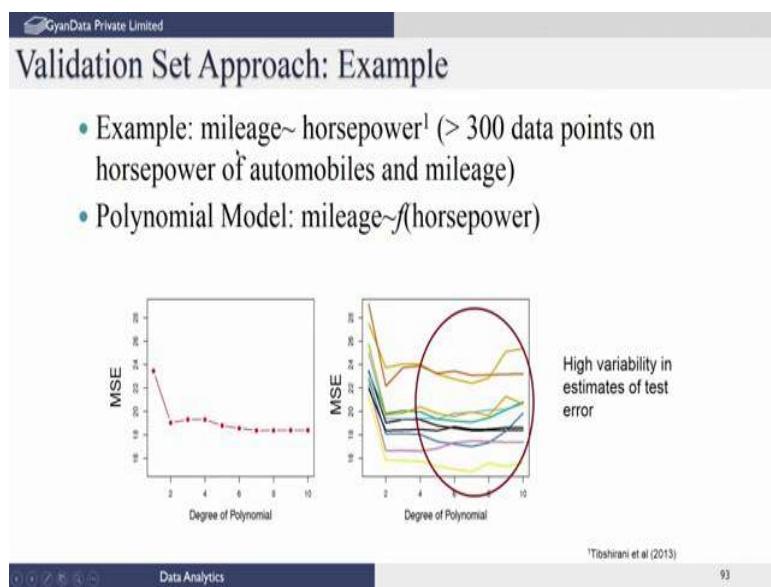
using what is called K fold cross validation and bootstrapping and so on.

So, this is what we will do if you do not have enough data. We will first look at the case of where we have sufficient number of samples. So, essentially we have n samples and you can divide it to a training set consisting of n-t samples and the remaining samples you actually use for validation. This is the hold out sample as we call it.

So, you build the model using only the training set and after you have built the model you test it on the validation set to find the mean square error. Do this for every choice of the parameter. So, if you have for example, a polynomial model, you will first see whether a linear model is good then you will check a quadratic model and a cubic model and so on. You keep increasing the degree of the polynomial and for each case, build the model using the training set and see how the MSE of that particular model is on the validation set and plot this MSE on the validation set as a function of the degree of the polynomial.

So, this is what we are going to do for one of the examples and then see how we pick the optimal number of parameters. So, here is a case example of mileage of some auto automobiles and the horse power of the engine.

(Refer Slide Time: 10:53)



So, essentially this particular data set contains 300 data points. Actually this is sufficiently large, but we are going to assume this is not large enough and we use the validation and cross validation approach on this data set. So, we have 300 data points or more of automobiles, different types of automobiles, for which the horsepower and the

mileage is given. We are going to fit a polynomial or a non-linear model between mileage and horsepower, yeah of course we can also try a linear model, but a polynomial model means you can also try quadratic and cubic models and so on, so forth. So, this is what we are going to illustrate.

So, as we increase the degree of the polynomial, here what we have shown is the mean squared error on the validation set. So, suppose we take 70 percent of this for training and the remaining 30 percent or 100 data points or so for validation and we look at the mean squared error on the validation set for different choices of the polynomial order. In this case, the first polynomial degree is 1 implies we are taking a linear model and for 2 implies we are fitting a quadratic model, for 3 implies a cubic model and a quartic model and so on, so forth and we have tried polynomial up to degree 10 and we have shown how the mean squared error on the validation data set is as you increase the polynomial order.

You can see very clearly that the value reaches more or less or minimum at 2 and afterwards it does not significantly change. Typically this should actually start increasing but in most experimental data sets you will find that the mean squared error on the validation set flattens out and does not significantly decrease beyond the point.

So, you can choose the optimal order degree of the polynomial to fit if this particular case as 2 that is a quadratic model fits this data very well. That is what you actually conclude from this particular cross validation mechanism. Of course on the right hand side we have shown plots for different choices of the training set. For example, if you choose 200 data points out of this randomly and perform regression, polynomial regression, for different polynomial order degree and plot the MSE, you will get let us say one curve in this case let us say the yellow curve you get here.

Similarly, if you take another random set and do it, you will get another curve. So, these different curves correspond to different random samples taken from this 300 thing as training and the remaining is used as testing. You can see that as you increase the degree of the polynomial, the variability or the estimates or the range of the estimates is very very large.

So, it indicates that if you overfit, you will get a very high variability whereas on the other hand if you choose order of the polynomial 1 or 2 you will find that the variability is not that significant comparatively. So, typically if you overfit the model, you will find high variability in your estimates that you obtain or the mean squared error values that you obtain. All this is good if you have a large data set.

(Refer Slide Time: 14:10)

The slide has a dark blue header bar with the GyanData Private Limited logo. The main title 'Sampling for small data sets' is in bold black font on a light gray background. Below the title is a bulleted list of four items:

- Validation of models by repeatedly drawing random samples from a training set
  - Validation set (random sampling)
  - K-fold cross validation
  - Bootstrap
- Objective: Predict the performance of model(s) on the validation/test sets (drawn from training data)
- Resampling methods useful for data scarce situations

What happens when you have extremely small data set and you cannot divide it into training and validation set. You do not have sufficient samples for training. Typically, you need reasonable number of samples in the training set to build the model. Therefore, in this case we cannot set apart or divide it into a 70, 30, what I call, division and therefore you have to do some other strategies. So, these strategies or what are called cross validation using a k fold cross validation or a bootstrap. That we will see.

Here again, we will predict the performance of the model on the validation set, but the validation set is not separated from the training set precisely. But on the other hand, it is drawn from the training set and we will see how we do this. So, these methods k fold cross validation is useful when we have very few samples for training.

(Refer Slide Time: 15:08)

GyanData Private Limited

## Leave-one-out-cross-validation (LOOCV)

- Build model using  $(n-1)$  samples and predict the response ( $y_i$ ) for the remaining sample

$$CV_1 = \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i^T \hat{\boldsymbol{\beta}}^{(1)})^2$$

95

So, I will first start with, leave one out cross validation or what is called LOOCV. In this case, we have  $n$  samples as I said  $n$  is not very large may be 20 or 30 samples we have for training. So, what we will do is you first leave out the first sample and use the remaining samples for building your model. That means, you will use samples 2, 3, 4 up to  $n$  to build your model and once you have built the model you test the performance of the model or predict for this sample that you have left out and you will get an MSE for the sample.

And similarly, in the next round what you do is, leave out the second sample and choose all the remaining for training and then use that model for predicting on the sample that is left out. So, in every time, you build a model by leaving out one sample from this list of  $n$  samples and predict the performance of the model on the left out sample.

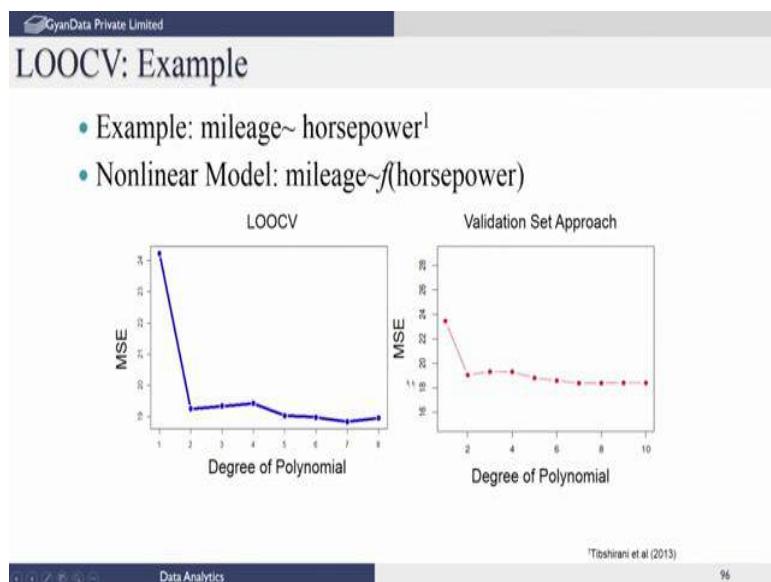
This you will do for every choice of the model parameter. For example, if you are building a non-linear regression model, you will build an regression model using let us say only  $\beta_0$  and  $\beta$  one the linear term and predict the MSE on this left out sample. You will also build a model using the same training set, quadratic model, and predict it on this and so on, so forth. So, that you will get a MSE for the left out sample for all choices of the parameters; that you want to try out.

Do this with the second sample being left out and the third sample being left out in turn. So that every sample has a chance of being in the validation set and also be being part of the training set in the other cases. So, once you have done this, for a particular choice of the model parameters, let us say you have building a linear model. You find out

the sum squared value of the prediction errors on the left out sample over all the samples. for example, you would have got a MSE for this, MSE for this, MSE for this.

When the first sample, second sample and third sample was left out that you are cumulating it here and taking the average of all that. This you do repeatedly for every choice of the parameters in the model. For example, the linear the quadratic the cubic and so on so forth and you get the mean squared error or cross validation error for different values of the parameters which you can plot.

(Refer Slide Time: 17:36)



Here again we have shown the mean squared error for different choice of the degree of the polynomial for the same data set. In this case, we have used left Leave One Out Cross Validation strategy. That means, if we have 300 samples we left out one sample, built the model using 299 samples predicted on the sample that is left out do this in turn, average over all of this for every choice of these parameter, degree of the polynomial and mean squared error we have plotted.

Again we see that the MSE on the cross validation leave one out cross validation reaches more or less the minimum for a degree of the polynomial is equal to 2, after which it just keeps remains more or less at. So, the optimal in this case is also indicated as a second order polynomial is best for this particular example.

(Refer Slide Time: 18:24)

GyanData Private Limited

## LOOCV

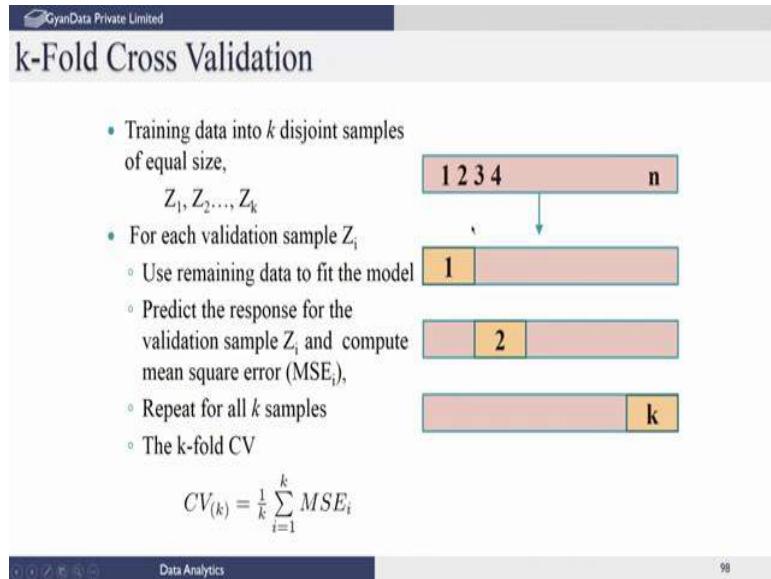
- Leave-one-out-cross-validation (LOOCV)
- Advantages
  - Far less bias comparison to the validation set approach
  - Training set contains  $(n-1)$  observations each iteration
  - Yield the same results
    - No randomness in the training/validation set splits
  - Does not overestimate the test error rate as much as the validation set approach
- Disadvantages
  - Expensive to implement due to fitting happens  $n$  times
  - It may select a model of excessive size (more variables) than the optimal model

97

So, Leave One Out Cross Validation has an advantage as compared to the valuation set approach, when to, we can show that it does not overestimate the test error rate as much as the validation set approach. It is comparatively expensive to implement because you are building the model  $n$  times, one for each sample being left out and you have to repeat this for all choice of the hyper para model parameter. For example, we have to do this for the linear model the quadratic model the cubic model and so on so forth.

So, you have not only have to do this  $n$  times, but you have to do this  $n$  times for every choice of the number of parameters of the model. So, it is quite a lot of computation that it takes. In general it may actually sometimes fit a model which is slightly more than the optimal model by not always, but sometimes it is also possible that the Leave One Out Cross Validation procedure over fits the model

(Refer Slide Time: 19:27)



We can also do what is called the  $k$  fold cross validation where here instead of leaving one out, we first divide the entire training set into  $k$  folds or  $K$  groups. So, let us say the first group contains, let us say, the first 4 data samples the second group contains the next 4 and so on, so forth and we have divided this entire  $n$  samples into  $k$  groups. Now instead of leaving one out, we will leave one group out. So, for example, in this first case we will leave the first 4 samples that belong to group one and use the remaining samples and build a model for whatever choice of the parameters we have used, let us say we are building a linear model.

We will use the remaining groups, build the linear model and then predict for the set of samples in group 1 that was left out and compute the MSE for this group. Similarly in the next round, what we will do is leave group 2 out, build the model let us say the linear model that we are building with the remaining groups and then find the prediction error for group 2 and so on, so forth, until we find the prediction error for group  $k$  and then we average over all these groups.

So, the MSE in this case for all groups, where there are  $k$  groups, and  $1/k$  that will be the cross validation error for leave this  $k$  fold cross validation. Now, you can you have to repeat this for every choice of the parameter, you have done this for the linear model you have to do this for the quadratic model cubic model and so on, so forth and then you can plot this cross validation error for leave or for this  $k$  fold cross validation.

(Refer Slide Time: 21:00)

## k-fold Validation

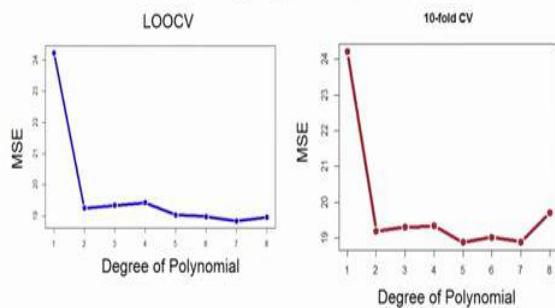
- For  $k=n$ , Leave-one-out-cross-validation (LOOCV)
- In practice,  $k=5$  or  $10$  is taken,
- Less computation cost
- For computationally intensive learning methods
  - LOOCV fits the model  $n$  times
  - $k$ -fold CV fits the model  $k$  times

Notice that if  $k = n$ , you are essentially going back to Leave One Out Cross Validation. In practice you can choose the number of groups is equal to either 5 or 10 and do a 10 fold cross validation or 5 fold cross validation. This is obviously less expensive computationally as compared to Leave One Out Cross Validation and as you see that leave one out cross validation will do a model fitting  $n$  times for every choice of the parameter whereas  $k$  fold cross validation will do the model building  $k$  times for every choice of the parameter.

(Refer Slide Time: 21:39)

## k-fold CV: Example

- Example: mileage~ horsepower<sup>1</sup>
- Nonlinear Model: mileage~f(horsepower)



Again we have illustrated this  $k$  fold cross validation for this mile auto data, again we plot the MSE for different degrees of the

polynomial. We have used a 10 fold cross validation and we are plotting this error.

And we will see that here also the minimal error occurs at 2 showing that a quadratic model is probably best for this particular data after which the error actually essentially flattens out. So, cross validation is an important method or approach for finding the optimal number of parameters of a model. This happens in clustering, this will happen in non-linear model fitting and principal component analysis and so on and it is useful. Later on, you will see in the clustering lectures, the use of cross validation for determining the optimal number of clusters.

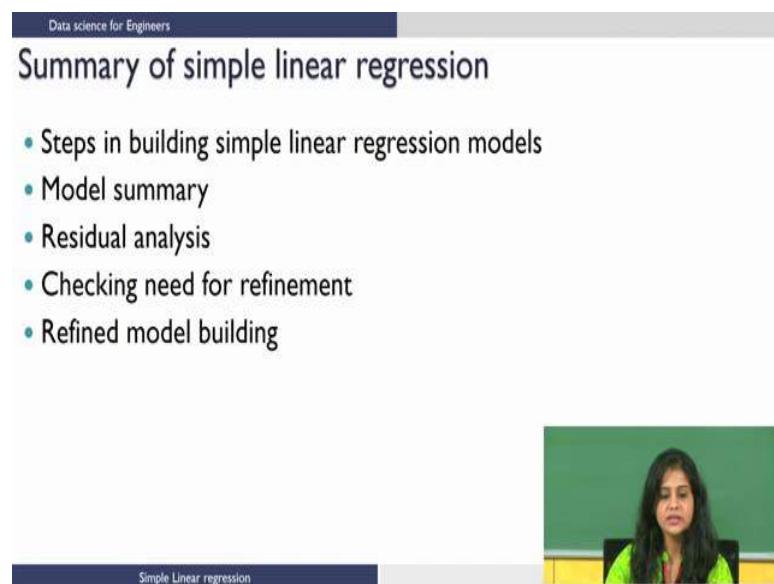
Thank you.

**Data science for Engineers**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture – 40**  
**Multiple Linear Regression Model Building and Selection**

Welcome to the lecture on implementation of multiple linear regression to summarize from the previous lecture.

(Refer Slide Time: 00:23)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. The main title 'Summary of simple linear regression' is centered in a large, bold, black font. Below the title is a bulleted list of six items, each preceded by a teal dot. The list includes: 'Steps in building simple linear regression models', 'Model summary', 'Residual analysis', 'Checking need for refinement', and 'Refined model building'. At the bottom of the slide, there is a dark blue footer bar with the text 'Simple Linear regression' in white. To the right of the slide, there is a small video thumbnail showing a woman with long dark hair, wearing a green top, sitting at a desk and speaking. The background behind her is a green chalkboard.

- Steps in building simple linear regression models
- Model summary
- Residual analysis
- Checking need for refinement
- Refined model building

We looked at steps in building a simple linear regression model where we looked at how to regress an independent variable with a dependent variable. As a part of this we also looked at how to assess the model that we have built and under that we looked at how to interpret the model summary and identify the significant variables. How to do residual analysis, how to check if the model needs refinement and we built a refined model.

(Refer Slide Time: 00:57)

Data science for Engineers

## In this lecture

- Multiple linear regression
  - Build linear model with one dependent and multiple independent variables
  - Look at summary of the model to discard insignificant variable
  - Model selection

Simple Linear regression



In this lecture we are going to extend all of this to multiple independent variable so, it is called multiple linear regression and in this we are going to build linear model with one dependent and multiple independent variables. We are also going to look at the model summary and identify the insignificant variables and discard them and rebuild the model. We will also look at how to identify the subset of variables to build the model, this is called model selection.

(Refer Slide Time: 01:20)

Data science for Engineers

## Loading data

- Dataset 'nyc' is given in ".csv" format
- To load data from the file the function used is `read.csv( )`

Simple Linear regression



So, let us start by loading the data so, the data set ‘nyc’ is given to you in a “csv” format and to load the dataset we are going to use the function read dot csv.

(Refer Slide Time: 01:30)

Data science for Engineers

## read.csv()

Reads a file in table format and creates a data frame from it

SYNTAX

```
read.csv(file, row.names=1)
```

|           |                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file      | the name of the file which the data are to be read from. Each row of the table appears as one line of the file.                                                                                                                            |
| row.names | a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names. |

Simple Linear regression



So, the inputs for the function read dot csv it is similar to what we saw in the previous lecture for read dot delim. So, read dot csv reads the file in the table format and creates a data frame from it. So, the syntax is read dot csv and the inputs to the function are file and row names. So, le is the name of the file from which you want to read the data and row names is the vector giving the actual row names, could also be a single number.

(Refer Slide Time: 02:00)

Data science for Engineers

## Loading data

- Assuming that ‘nyc.csv’ is in your current working directory

```
nyc <- read.csv("nyc.csv")
```

- The data is saved into a data frame ‘nyc’

Simple Linear regression



So, let us see how to load the data now so, assuming ‘nyc.csv’ is in your current working directory the command is read dot csv followed by the name of the le in double quotes. Now, once this command is executed it will create an object nyc which is a data frame. Now, let us see how to view the data.

(Refer Slide Time: 02:23)

Data science for Engineers

## Viewing data

- `View(nyc)` will display the dataframe in a tabular format

|   | Price | Food | Decor | Service | East |
|---|-------|------|-------|---------|------|
| 1 | 43    | 22   | 18    | 20      | 0    |
| 2 | 32    | 20   | 19    | 19      | 0    |
| 3 | 34    | 21   | 13    | 18      | 0    |
| 4 | 41    | 20   | 20    | 17      | 0    |

- `head(nyc)` and `tail(nyc)` will display the first and last six rows from the dataframe

Simple Linear regression



Now view of nyc will display the data frame in a tabular format. There is a small snippet below which shows you how the output looks. So, I have price, food, decor, service and east as the 5 variables. So, say suppose if your data is really huge and you do not want to view the entire data then we can use head or tail function. So, head will give you the first 6 rows from a data frame and tail will give you the last 6 rows from the data frame.

So, now, let us look at the description of the data set we have already loaded it and we viewed it, but we do not know yet what the description is.

(Refer Slide Time: 02:57)

Data science for Engineers

## Description of dataset

Menu pricing in restaurants of NYC

$y$  : Price of dinner  
 $x_1$ : Customer rating of the food (Food)  
 $x_2$ : Customer rating of the décor (Décor)  
 $x_3$ : Customer rating of the service (Service)  
 $x_4$ : If the restaurant is east or west (East)

Objective: Build a linear model



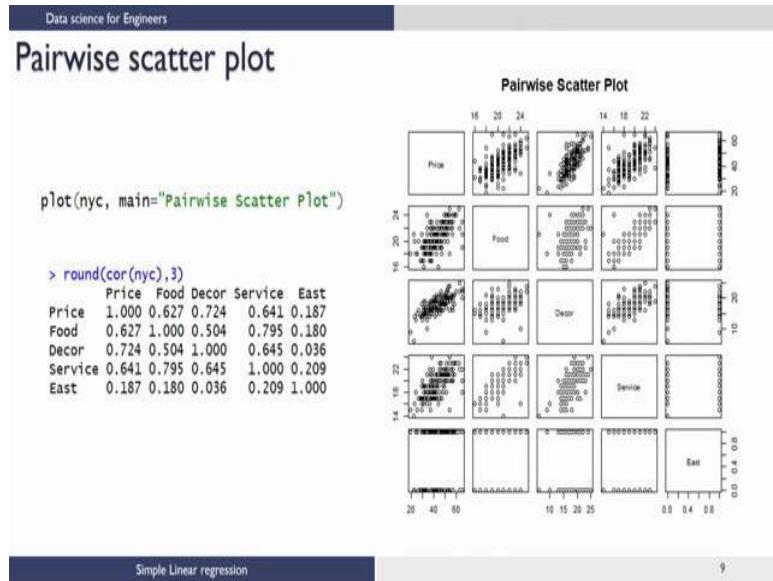
Simple Linear regression

So, the data is about menu pricing in restaurants of New York City. So,  $y$  which is my dependent variable is the price of the dinner, there are 4 other independent variables. So, I have food which is one of the independent variables it, is the customer rating of the food then I have decor which is the customer rating of decor, then I have service which is the customer rating of the service and east.

So, east is whether the restaurant is located on the east or west side of the city. So, now, our objective is to build a linear model with  $y$  which is price and with all the other 4 independent variables. Before we go on building a model let us say if our data exhibits some interdependency between the variables. So, for me to do that I am going to use a "pair wise scatter plot." So, I am going to use same function plot which we have earlier used.

Now, since I have multiple variable I am going to give the data frame as my input and I am just giving a heading as pair wise scatter plot.

(Refer Slide Time: 04:06)



On my right this is the output you will get so, we can see that all the variables are mentioned across the diagonals. So, when one moves from left to right the variables on my left will be in the y axis and the variables above or below will be on the x axis. So, let us take the first row for instance. So, I have price on the left. So, price is in the y and I have food below. So, food becomes the x axis.

Now, this is the plot for price versus food similarly I have price versus decor and price versus service and price versus east. I am going to the next row which is food on the y axis. So, if you take food versus decor the data is randomly scattered so, it does not show any correlation patterns, but whereas, if you see for food versus service you see strong patterns being exhibited here. So, let us see what the correlation is as such for all of these. So, correlation is a function and Professor Shankar has told you how it is computed.

So, cor is the function in R. I need to give the dataset with all the variables now round tells you to how many decimal points you want round off the number to. So, if I give round and I am giving the input as my correlation function and if I am saying 3 it means round of the number to 3 decimal places. So, let us see how to interpret the output. So, the correlation for price versus price will always be 1.

So, let us look at food and decor so, correlation between food and decor is 0.5 which is pretty low, but whereas, if you look at food and service it is almost equal to 0.8 which is quite high. So, we can see that food and service are correlated, but one of them can be dropped while building a final model. So, as we go along let us see which of the two we have to drop.

(Refer Slide Time: 06:25)

## Model Building

Now, let us go on to model building.

(Refer Slide Time: 06:28)

Data science for Engineers

### Building multiple linear regression model

- Dependent variable ( $y$ ) depends on  $p$  independent variables  $x_i, i=1,2..p$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_p x_p + \epsilon$$

- For  $i^{\text{th}}$  observation,

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{1,i} + \hat{\beta}_2 x_{2,i} + \dots + \hat{\beta}_p x_{p,i} + \epsilon_i$$



Simple Linear regression

So, like I earlier said, my dependent variable is only one here mean which is denoted by  $y$ . I have several independent variables which are denoted by  $x_i$  and  $i$  code ranges from 1 to  $p$ , where  $p$  is the total number of independent variables. Now let us see how to write this equation with multiple independent variables. Again I have  $\hat{y}$  which is the predicted value now I have  $\beta_0$  which is the intercept then I have  $\beta_1 x_1 + \beta_2 x_2$  so on and so forth up to  $\beta_p x_p$ . So,  $\beta_0$  is the intercept and  $\beta_1$

hat  $\beta_2$  hat etcetera are the slopes.

So,  $\varepsilon$  is the error. So, if you could recall from your earlier lectures in OLS, the assumption is that, so, error is present only in the measurement of dependent variable and not on the independent variable. So, independent variables are free of errors whereas, there is always some error present in the measurement of  $y$ . So, this  $\varepsilon$  is an unknown quantity which has 0 mean and some variance, now for any  $i$  th observation this is how my equation is written.

(Refer Slide Time: 07:42)

Data science for Engineers

## Building multiple linear regression model

- Building multiple linear model using the function `lm()`
- Syntax: `lm(formula,data)`  
`lm(dependent var~indep.var1+ indep.var2)`  
`nycmod_1 <- lm(Price~Food+Decor+Service+East,data = nyc)`  
or  
`nycmod_1<-lm(Price~,data=nyc)`

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p + \epsilon$$


Simple Linear regression

So, now, let us go and build a model. So, the function to build a multiple linear model is same as what we used in the univariate case. Here also I am going to use `lm` now again the syntax is `lm` and there are 2 input parameters formula and data. Now the syntax is slightly different compared to the univariate case. So, I have my dependent variable here then I have a tilde sign and how many ever independent variables I have I am going to separate them with a + sign. Say for instance I have 2 independent variables in my data. So, I am regressing the dependent variable with 2 independent variables so, the 2 independent variables have to be separated by a + sign.

So, now, let us see how to do it for our data `nyc`. So, again I have `lm` so, I am regressing price with all the 4 input variables which is food, decor, service and east and I am taking these variables from the data `nyc`. So, you can either separate the independent variables by a + sign. So now, if you want to say regress price with all the 4 inputs, there is another way you can write the same command. So, I say regress price and then I give a tilde sign and then I say a dot. So, this means regress price with all the input variables from the data `nyc` So, if you are going to give all the input variables for regression then you can go with this,

but if you have a subset of variables that you want to build a model with, then you can specify the variables separated by a + sign. So, just to reiterate this is the form of my equation. So, now, let us go and see how to interpret the summary. So, after having built this model I am going to look at the summary of it.

(Refer Slide Time: 09:30)



```
Data science for Engineers
Model summary

nycmod_1 <- lm(Price~Food+Decor+Service+East,data = nyc)
summary(nycmod_1)

Call:
lm(formula = Price ~ Food + Decor + Service + East, data = nyc)

Residuals:
 Min 1Q Median 3Q Max
-14.0465 -3.8837 0.0373 3.3942 17.7491

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -24.023800 4.708359 -5.102 9.24e-07 ***
Food 1.538120 0.368951 4.169 4.96e-05 ***
Decor 1.910087 0.217005 8.802 1.87e-15 ***
Service -0.002727 0.396232 -0.007 0.9945
East 2.068050 0.946739 2.184 0.0304 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.738 on 163 degrees of freedom
Multiple R-squared: 0.6279, Adjusted R-squared: 0.6187
F-statistic: 68.76 on 4 and 163 DF, p-value: < 2.2e-16
```

So, this snippet gives you a just at the summary. So, if you could recall in the first lecture of simple linear regression, we looked at what each of this line here means in depth. So, we have the formula in the first line we have the residuals and the 5 point summary of them ,then we look in at the coefficients. So, here we say that intercept, food, decor, service and east and these are the coefficients for these variables.

So, for each of these coefficients, I am given an estimate value some variance associated with it a t value which is the ratio of estimate by the standard error and some probability value. So, if you look at the p value. So, the p value for intercept is very low compared to our significance level which is 0.05. So, this tells you that intercept is one of the important terms that have to be included in the model. The same goes with p value for food and decor they are also less than 0.05 and so we need to retain them and the stars tell you how significantly different are they from 0.

Whereas, if you look at the p value of service, it is 0.9945 which is really high compared to our significance level. So, this tells you that this term, service, is not important and if you look at the estimated value it is very very close to 0. So, this tells you that service is not an important term in explaining the price.

So, now, if you see the p value for east, though it is not very very low compared to food and decor it is though it does not have a p value

which is very low as that compared to food and decor, it is still OK and the significance star is only one which tells you that look if I have a significance level of say 0.025 or 0.01 then I can reject this term, but till then I can always keep it.

So, let us look at the r squared value. The r squared value is 0.628 and the adjusted r square is 0.619 and the f statistic value is really high which is 68.76. So, this tells you that compared to the reduced models which are the only intercept my full model is performing better and I should retain it. So, now, that we know service is not significant, let us build a new model dropping service.

(Refer Slide Time: 12:11)

```

Data science for Engineers

New model dropping Service

nycmod_2 <- lm(Price~Food+Decor+East,data = nyc)
summary(nycmod_2)

Call:
lm(formula = Price ~ Food + Decor + East, data = nyc)

Residuals:
 Min 1Q Median 3Q Max
-14.0451 -3.8809 0.0389 3.3918 17.7557

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -24.0269 4.6727 -5.142 7.67e-07 ***
Food 1.5363 0.2632 5.838 2.76e-08 ***
Decor 1.9094 0.1900 10.049 < 2e-16 ***
East 2.0670 0.9318 2.218 0.0279 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

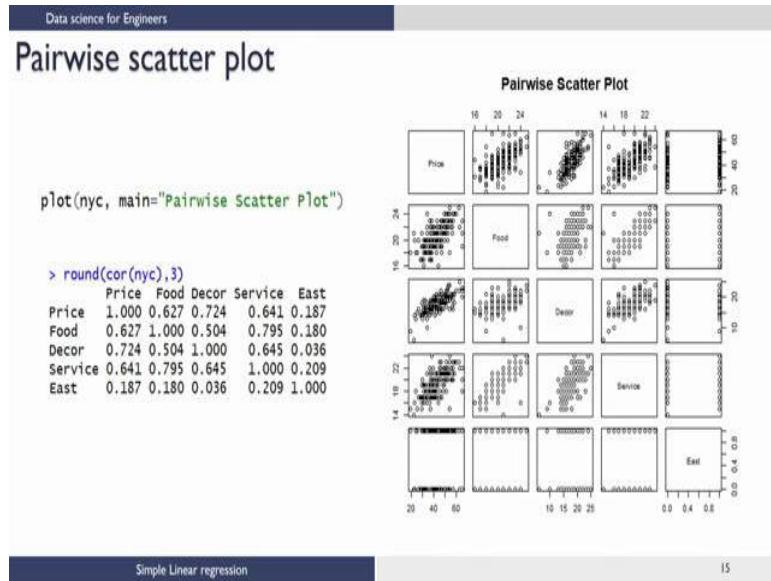
Residual standard error: 5.72 on 164 degrees of freedom
Multiple R-squared: 0.6279, Adjusted R-squared: 0.6211
F-statistic: 92.24 on 3 and 164 DF, p-value: < 2.2e-16

```

So, I have dropped service and I have built a new model and I am calling it nycmod\_2. So, let us jump on to the coefficient section. So, the estimates are not drastically different before and after removing the service variable. So, this tells you that service is not very important. So, again if you look at the p value it tells you that these variables are very significant and if you look at the r squared value here down.

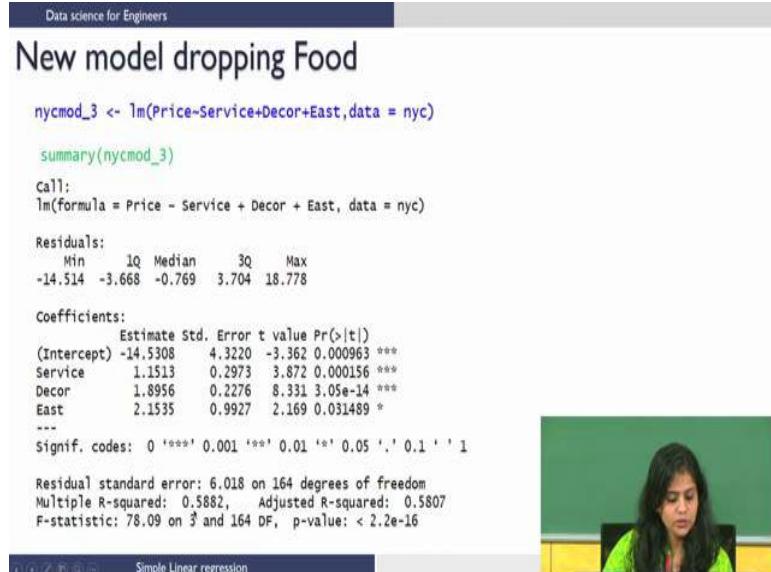
So, the r squared value before and after removing service is not changed much this itself is an indicator that service is not helping us in explaining the variation in price. The adjusted r square has changed a bit and that is only because we have removed one variable and the degrees of freedom change. The f statistic again is really really high telling you that full model with food, decor and east is performing better compared to your reduced model with only the intercept.

(Refer Slide Time: 13:15)



If you recall from the scatter plot, we saw there was a high correlation between food and service. So, now, we built a model dropping service, let us now retain service and build a model dropping food. So, I have dropped food from here. So, let us take a look at this summary.

(Refer Slide Time: 13:28)



If you take a look at this summary though the p value tells you that all the variables are significant, if you look at the r squared value it has dropped from 0.628 to 0.588 which is a huge decrease and even the

adjusted r square has decreased. So, this tells you that service is less important and food is explaining the price in a much better sense than service.

So, the r squared value and the scatter plots tell us to go ahead with the linear model where we still need to verify the assumptions we make on the errors using residual analysis. So, this task we are going to leave it to you as an exercise you can do it and verify these assumptions.

Thank you.

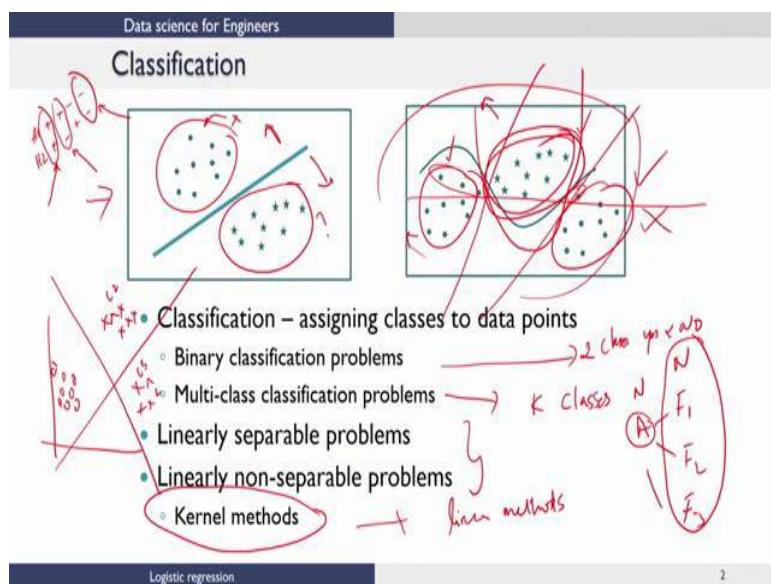
**Data science for Engineers**  
**Prof. Ragunathan Rengasamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture – 41**  
**Classification**

Let us continue our lectures on algorithms for data science, we will now see some algorithms that are used in what we call as the classification problems. I will first start by discussing classification problems again. We have done this before, but I thought I will come back to this for this part of the lectures and then I will tell you the type of classification problems quickly and then describe some characteristic that we look for in these classification problems and then I will teach three techniques which could be used in solving classification problems.

In general many of the techniques that I used in classification can also be modified or used for function approximation problems. So, this is something that you should keep in mind, similarly techniques used for function approximation problems can also be modified and used for classification problems. None the less in this series of lectures we will look at these algorithms and then give them a classification flavour and that would come through both the way we describe the algorithm and also the case studies that are used with the algorithms.

(Refer Slide Time: 01:50)



So, let us look at what classification means. So, we have described this before if I am given data that is labelled to different classes that is what I start with, then if I am able to develop an algorithm which will be able to distinguish these classes. And how do you know that this algorithm distinguishes these classes. Whenever a new data point comes if I send it through my algorithm and if I originally had K different labels the algorithm should label the new point into one of the K groups.

So, that is typically what is called as classification problem. So, pictorially we represent here, here we say, here is a group of data, here is a group of data. Now, if I were to derive a classifier, then line like this could be a classifier and remember we have seen this in linear algebra before, I have 2 half spaces and I could say this half space is class 1 and this half space is class 2. Now, you noticed that while I have data points only in a small region this classifier has derived an unbounded classification region. You could come up with classifiers which are bounded also we will discuss that later.

But here now if I get a new point if I get a point like this, then I would like the classifier to say that this point most likely belongs to the class which is denoted by star points here and that is what would happen and similarly if I have a point here I would like that to be classified to this class and so on. Now, we can think of two types of problems. The simpler type of classification problem is what is called the binary classification problem. Basically binary classification problems are where there are 2 classes, yes and no.

So, examples are for example, if you have data for a process or an equipment and then you might want to simply classify this data as belonging to normal behaviour of the equipment or abnormal behaviour of the equipment. So, that is just binary. So, if I get a new data I want to say from this data if the equipment is working properly or it is not working properly. Another example would be if let us say you get some tissue sample and you characterize that sample through certain means and then using those characteristics you want to classify this as a cancerous or a non cancerous sample. So, that is another binary classification example.

Now, complicated version of this problem is what we call as a multiclass classification problem where I have labels from several different classes. So, here I have just 2, but in a general case in a multi class problem, I might have K classes. A classic example again going back to the equipment example that we just described, instead of saying if the equipment is just normal or abnormal. If we could actually further resolve this abnormality into several different fault classes, let us say fault 1, fault 2, fault 3 then if you put all of this together normal fault 1, fault 2, fault 3, now you have a 4 class problem.

So, if I have annotated data where the data is labelled as being normal or as being collected when fault  $F_1$  occurred or as having been collected when fault  $F_2$  occurs, fault  $F_3$  occurs and so on, then whenever a new data point comes in we could label it as normal in which case we do not have to do anything or if we could label it as one of these fault classes then we could take appropriate action based on what fault class it belongs to. Now from a classification view point, the complexity of the problem typically depends on how the data is organized.

Now, if the data is organized, let us talk about just binary classification problem and many of these ideas translate to multi class classification problems. In a binary classification problem if the data is organized like it is shown in this picture here. We call this data as linearly separable where I could use hyper plane to separate this data into 2 sides of the hyper plane or 2 half spaces and that gives me perfect classification.

So, these are types of problems which are called linearly separable problems. So, in cases where you are looking at linearly separable problems the classification problem then becomes one of identifying the hyper plane that would classify the data.

So, these I would call as simpler problems in binary classification. However, I also have a picture on the right hand side this also turns out to be a binary classification problem. However, if you look at this, this data and this data both belong to class 1 and this data belongs to class 2. Now however you try to draw a hyper plane.

So, if we were to draw a hyper plane here and then say this is all class 1 and this is class 2 then these points are classified correctly, these points are classified correctly and these points are poorly classified or misclassified. Now, if I were to come out similarly with a hyper plane like this you will see similar arguments where these are points that will be poorly classified.

So, whatever you do if you try to come up with something like this then, these are points that would be misclassified. So, there is no way in which I can simply generate a hyper plane that would classify this data into 2 regions. However, this does not mean this is not a solvable problem it only means that this problem is not linearly separable or what I have called here as linearly non separable problems.

So, you need to come up with not a hyper plane, but very simply in layman terms curved surfaces and here for example, if you were able to generate a curve like this then you could use that as a decision function and then say on one side of a curve I have data point belonging to class 2 and on the other side I have data points belonging to class 1.

So, in general when we look at classification problems we are we look at whether they are linearly separable or not separable and from

data science view point this problem right here becomes lot harder because when we look at the decision function in a linearly separable problem we know the functional form, it is a hyper plane, and we are simply going to look for that in the binary classification case.

However, when you look at non-linear decision boundaries, there are many many functional forms that you can look at and then see which one holds. For example, one could may be come up with something like this or one could may be come up with things where I just do something like this and so, on.

So, there are many many possibilities. Now which of these possibilities would you use is something that the algorithm by itself has to figure out. So, since there are many many possibilities these become harder problems to solve. So, all of this we described for binary classification problems. Many of these ideas also translate to multi class problems. For example, if you take let us say, I have data from 3 classes like this here. So, these are 3 classes. Now if I want to separate these 3 classes and then ask myself if I can separate this to through linear methods.

Now it is slightly different from the binary classification problem because we needed only one decision function and based on one decision function we could say whether a point belongs to class 1 or class 2. In multi class problems you could come up with more decision functions and more decision functions would mean more hyper planes and then you can use some logic after that to be able to identify a data point as belonging to a particular class.

So, when I have something like this here let us say this is class 1, this is class 2 and this is class 3 what I could possibly do is the following. I could do hyper plane like this and a hyper plane like this. Now then I have these 2 hyper planes, then I have basically 4 combinations that are possible. So, for example, if I take hyper plane 1, hyper plane 2, as the 2 decision functions, then I could generate 4 regions. For example, I could generate + +, + -, - +, - -. So, you know that for a particular hyper plane you have 2 half spaces, a positive half space and a negative half space.

So, when I have something like this here then basically what it says is, the point is in the positive half space of both hyper plane one and hyper plane 2 and when I have a point like this, this says the point is in the positive half space of hyper plane 1 and the negative half space of hyper plane 2 and in this case you would say it is in the negative half space of both the hyper planes.

So, now, you see that when we go to multi class problems if you were to use more than one hyper plane then depending on the hyper planes you get a certain number of possibilities. So, in this case when I

use this 2 hyper planes I got basically 4 spaces as I show here. So, in this multi class problem which is I have 3 classes, if I could have data belonging to one class falling here data belonging to another class falling here and let us say the data belonging to the third class falling here for example.

Then I could use these 2 hyper planes and the corresponding decision functions to be able to classify this 3 class problem. So, when we describe multi class problems we look at more hyper planes and then depending on how we derive these hyper planes we could have these classes being separated in terms of half spaces or a combination of half spaces.

So, this is another important idea that, that one should remember when we go to multi class problems. So, when we solve multi class problems, we can treat them directly as multi class problems or you could solve many binary classification problems and come up with a logic on the other side of these classification results to label the resultant to one of the multiple classes that you have.

So, these give you some basic conceptual ideas on how classification problems are solved particularly binary classification problems and multi class classification problems, the key ideas that I said that we want to remember here are whether these problems are linearly separable or linearly non separable and in the linearly non separable problems there are multiple options one way to address the multiple options is through a beautiful idea called Kernel methods, where this notion of checking several non-linear surfaces can still be solved under certain conditions on the non-linear functions that we want to check using simple I would I am going to call it linear methods and I will explain this later in the course.

So, the idea here is that if you choose certain forms of non-linear functions which obey certain rules and those rules typically are called you know Kernel tricks. Then you could use whatever we use in terms of hyper planes, those ideas, for solving those class of problems.

So, Kernel methods are important when we have linearly non separable problems. So, with this I just wanted to give you a brief idea on the conceptual underpinnings of classification algorithms the math behind all of this is what we will try to teach at least some of it is what we will try to teach in this course and in more advance machine learning courses you will see the math behind all of this in much more detail.

So, as far as classification is concerned we are going to start with an algorithm called logistic regression. We will follow that up with k n n classifier then we will teach something called k means clustering, k means come under what are called clustering techniques. Now, typically you can use these clustering techniques in function approximation or classification problems and I am going to teach these

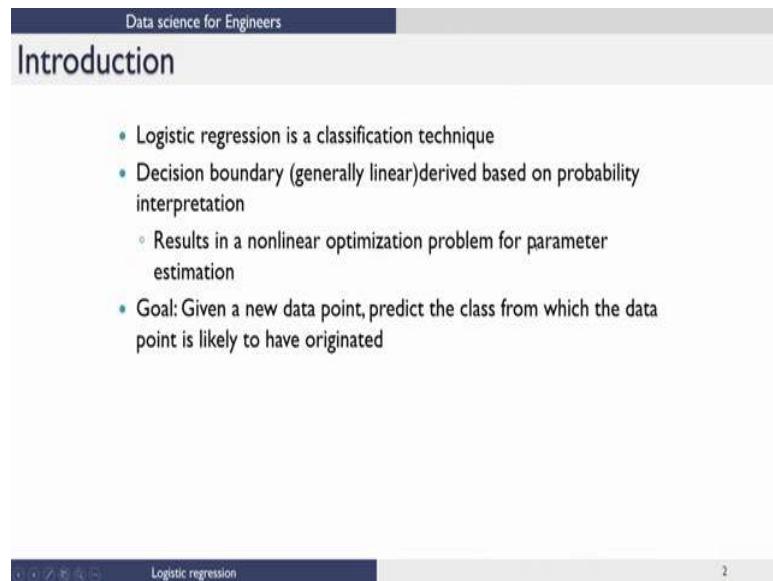
techniques and use case studies that give a distinct classification flavour to the results that we are going to see using these techniques. So, I will start logistic regression in the next lecture and I hope to see you then.

Thank you.

**Data science for Engineers**  
**Prof. Ragunathan Rengasamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 42**  
**Logistic regression**

(Refer Slide Time: 00:24)



The slide has a dark blue header bar with the text "Data science for Engineers". Below it is a light gray header section with the title "Introduction". The main content area is white with a dark blue footer bar containing icons and the text "Logistic regression".

- Logistic regression is a classification technique
- Decision boundary (generally linear) derived based on probability interpretation
  - Results in a nonlinear optimization problem for parameter estimation
- Goal: Given a new data point, predict the class from which the data point is likely to have originated

In this lecture, I will describe a technique called Logistic regression. Logistics regression is a classification technique which basically develops a linear boundary regions, based on certain probability interpretations, while in general we develop a linear decision boundaries, this technique can also be extended to develop non-linear boundaries using what is called polynomial logistic regression. For problems, where we are going to develop linear boundaries the solution still results in a non-linear optimization problem for parameter estimation, as we will see in this lecture. So, the goal of this technique is, given a new data point, I would like to predict the class from which this data point could have originated.

So, in that sense this is a classification technique, that is used in a wide variety of problems and it is surprisingly effective for a large class of problems.

(Refer Slide Time: 01:26)

Data science for Engineers

## Binary classification problem

- Classification is the task of identifying a category that a new observation belongs to based on the data with known categories
- When the number of categories is 2, it becomes a binary classification problem
- Binary classification is a simple "Yes" or "No" problem



Logistic regression

Just to recap the things that we have seen before, we have talked about binary classification problem before. Just to make sure that we recall some of the things that we have talked about before. We said classification is the task of identifying, what category a new data point, or an observation belongs to. There could be many categories to which the data could belong, but when the number of categories is 2, it is what we call as the binary classification problem. We can also think of binary classification problems as simple yes or no problems where, you either say something belongs to particular category, or no it does not belong to that category.

(Refer Slide Time: 02:15)

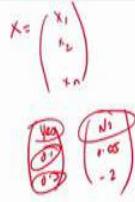
Data science for Engineers

## Input features

- Input features can be both qualitative and quantitative
- If the inputs are qualitative, then there has to be a systematic way of converting them to quantities
  - For example: A binary input like a "Yes" or "No" can be encoded as "1" and "0"
- Some data analytics approach can handle qualitative variables directly



Logistic regression



Now, whenever we talk about classification problems, we have described this before, we say a data is represented by many attributes,

$X_1$  to  $X_n$ . We can also call this as input features as shown in this slide. And these input features could be quantitative, or qualitative. Now quantitative features can be used as they are. However, if we have going to use a quantitative technique, but we want to use input features which are qualitative, then we should have some way of converting this qualitative features into quantitative values. One simple example is if I have a binary input like a yes or no for a feature. So, what do we mean by this. So, I could have yes let us say 0.1 0.3 and, another data point could be no 0.05 - 2 and so on.

So, you notice that these are quantitative numbers while these are qualitative features. Now, you could convert this all into quantitative features by coding yes as 1 and no as 0. So, then those also become number. This is very crude way of doing this, there might be much better ways of coding qualitative features into quantitative features and so on. You also have to remember that there are some data analytics approach, that can directly handle these qualitative features without a need to convert them into numbers and so on.

(Refer Slide Time: 04:03)

Data science for Engineers

## Linear classifier

- Decision function is linear
- Binary classification can be performed depending on the side of the half-plane that the data falls in
- We saw this before in the linear algebra module
- However, simply guessing "yes" or "no" is pretty crude
- Can we do something better using probabilities ?

Logistic regression

Raghunatha

So, you should keep in mind that that is also possible. Now that we have these feature, we go back to our pictorial understanding of these things. Just for the sake of illustration, let us take this example, where we have this 2 dimensional data. So, here we would say  $X$  is  $x_1 x_2$ ; two variables let us say  $x_1$  is here  $x_2$  is here. So, it is organized into data like this. Now let us assume that all the circular data belong to one category and all the starred data, belong to another category. Notice that circled data would have certain  $x_1$  and certain  $x_2$  and, similarly starred data would have certain  $x_1$  and certain  $x_2$ . So, in other words all values of  $x_1$  and  $x_2$  such that the data is here, belongs to one class and, such that the data is here belongs to another class.

Now, if we were able to come up with hyper plane such as the one that is shown here, we learn from our linear algebra module that to one side of this hyper plane is half space, this side is a half space and, depending on the way the normal is defined, you would have positive value and a negative value to each side of the hyper plane. So, this is something that we have dealt with in detail, in one of the linear algebraic classes.

So, if you were to do this classification problem, then what you could say is if I get a data point somewhere here, I could say it belongs to whatever this class is here. So, let us call this for example, we could call this class 0 we could call this class 1 and, we would say whenever a data point falls to this side of the line, then it is class 0 and if a data point falls to this side of the line, we will say it is class 1 and so on. However, notice that any data point. So, whether it falls here, or it falls here we are going to say class 0, but intuitively you know that if this is really a true separation of these classes, then this is for sure going to belong to class 0, but as I go closer and closer to this line there is this uncertainty about, whether it belongs to this class, or this class because data is inherently noisy.

So, I could have a data point, which is true value here; however, because of noise it could slip to the other side and so on. So, as I come closer and closer to this then you know the probability, or the confidence with which I can say it belongs to particular class, can intuitively come down. So, simply saying yes this data point and, this data point belongs to class 0 is 1 answer, but that is pretty crude. So, the question that this logistic regression answers is can we do something better using probabilities. So, I would like to say that the probability that this belongs to class 1 is much higher than this, because it is far away from the decision boundary. So, how do we do this, is the question that logistics regression addresses.

(Refer Slide Time: 07:44)

## Output

- Why model probabilities ?
  - The probability of a "Yes" or "No" gives a better understanding of the sample's membership to a particular category
  - Estimating the binary outputs from the probabilities is straight forward through simple thresholding
  - How does one model this probability ?



Logistic regression

So, as I mentioned before the probability of something being from a class, if we can answer that question, that is better than just saying yes or no answers, right.

So, one could say yes this belongs to class, a better nuanced answer could be that yes it belongs to class, but with a certain probability as the probability is higher, then you feel more confident about assigning that class to the data. On the other hand if you model through probabilities, we do not want to lose binary answer like yes or no also. So, if I have probabilities for something I can easily convert them to yes or no answers through some thresholding, which we will see in the logistics regression methodology when we describe that. So, while we do not lose the ability to categorically say, if a data belongs to a particular class or not by modelling this probability. On the other hand, we get a benefit of getting a nuanced answer, instead of just saying yes or no.

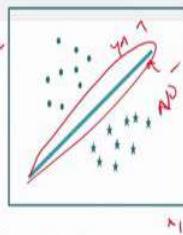
(Refer Slide Time: 08:52)

## Linear and log models

- Make  $p(x)$  a linear function of  $x$

$$p(x) = \beta_0 + \beta_1 X$$

$$\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$$



- This makes  $p(x)$  unbounded below 0 and above 1
- Might give nonsensical results making it difficult to interpret them as probabilities

- Make  $\log(p(x))$  a linear function of  $x$

$$\log(p(x)) = \beta_0 + \beta_1 X$$

- Bounded only on one side

So, the question then, is how does one model these probabilities. So, let us go back and look at the picture that we had before let us say this is  $x_1$  and  $x_2$ . Remember that this hyper plane, would typically have this form here the solution is written in the vector form. If I want to expand it in terms of  $x_1$  and  $x_2$ , what I could do is I could write this as  $\beta_0 + \beta_{11}x_1 + \beta_{12}x_2$ . So, this could be the equation of this line in this two dimensional space. Now one idea might be just to say this itself is a probability and then let us see what happens. The difficulty with this is, this  $p$  of  $x$  is not bounded because, it is just a linear function. Whereas you know that the probability has to be bounded between 0 and 1. So, we have to find some function which is bounded between 0 and 1. The reason why we are still talking about this linear function is because, this is the decision boundary.

So, what we are trying to do here is really instead of just looking at this decision boundary and then saying yes and no + and -, what we are trying to do is, we are trying to use this equation itself to come up with some probabilistic interpretation. That is the reason, why we are still sticking to this equation and trying to see if we can model probabilities as a function of this equation, which is the hyper plane.

So, you could think of something slightly different and, then say look instead of saying  $p$  of  $x$  is this let me say  $\log(p(x)) = \beta_0 + \beta_1 x$ . In this case you will notice that it is bounded only on one side. In other words, if I write  $\log(p(x)) = \beta_0 + \beta_1 x$ , I will ensure that  $p$  of  $x$  never becomes negative; however, on the positive side  $p$  of  $x$  can go to  $\infty$ . That again is a problem because we need to bound  $p$  of  $x$  between 0 and 1. So, this is an important thing to remember. So, it only bounds this on one side.

(Refer Slide Time: 11:16)

Data science for Engineers

## Sigmoid function

- Make  $p(x)$  a sigmoid function of  $x$

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

or  $\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$

- $p(x)$  bounded above by 1 and below by 0
- Good modeling choice for real life scenarios
- The LHS can be interpreted as the log of odds-ratio in the second equation

Logistic regression

So, is there something more sophisticated we can do? The next idea is to write  $p$  of  $X$  as what is called as sigmoidal function. The sigmoidal function has relevance in many areas. So, this is the function that is used in neural networks and other very interesting applications.

So, the sigmoid has an interesting form which is here. So, now, let us look at this form right here. I want you to notice two things number one is still we are trying to stick this hyperplane equation into the probability expression because, that is the decision surface. Remember intuitively somehow we are trying to convert that hyper plane into probability interpretation. So, that is the reason why we are still sticking to this  $\beta_0 + \beta_1 x$ . Now let us look at this equation and then see what happens.

So, if you take this argument  $\beta_0 + \beta_1 x$ . So, that argument depending on the value of  $X$  you take, could go all the way from  $-\infty$  to  $\infty$ . So, just take a single variable case if I if I write let us say  $\beta_0 + \beta_1$  just 1 variable  $X$  not a vector. Now if  $\beta_1$  is positive, if you take  $X$  to be a very very large value, this number will become very large, if  $\beta_1$  is negative, if you take  $X$  to be very very large value in the positive side this number will become  $-\infty$ . And similarly if  $\beta$  one takes the other values, you can correspondingly choose  $X$  to be positive or negative and then make this unbounded between  $-\infty$  to  $\infty$ .

So, we will see what happens to this function when  $\beta_0 + \beta_1 x$  is  $-\infty$ , you would get this to  $e$  power  $-\infty$  divided by  $1 + e$  power  $-\infty$ , or you can just think of this as - very large number. So, in that case when numerator will become 0 and, the denominator will become  $1 + 0$ . So, on the lower side this expression will be bounded by 0.

Now if you take  $\beta_0 + \beta_1 x$  to be a very large positive number, then the numerator will be a very very large positive number and the denominator will be  $1 + \text{that very large positive number}$ . So, this will be bounded by 1 on the upper side. So now, from the equation for the hyper plane, we have been able to come up with the definition of a probability, which makes sense, which is bounded between 0 and 1. So, it is an important idea to remember. By doing this what we are doing is the following. If we were not using this probability, all that we will do is we will look at this equation and whenever a new point comes in we will evaluate this  $\beta_0 + \beta_1 x$  and then based on whether it is positive or negative, we are going to say yes or no.

Now, what has happened is instead of that, this number is put back into this expression and depending on what value you get you get a probabilistic interpretation. That is the beauty of this idea here. You can rearrange this in this form and then say  $\log(p(X)) / (1 - p(X)) = \beta_0 + \beta_1 x$ . The reason why I show you this form is because the left hand side could be interpreted as log of odds ratio, which is an idea that is used in several places. So, that is the connection here.

(Refer Slide Time: 14:58)

Data science for Engineers

## Estimation of the parameters

- We find parameters in such a way that plugging these in the model equation should give the best possible classification for the inputs from both the classes
- This can be formalized by maximizing the following likelihood function

$$L(\beta_0, \beta_1) = \prod_{i=1}^n (p(x_i))^{y_i} (1 - p(x_i))^{(1-y_i)}$$

when  $x_i$  belongs to class 0,  $y_i = 0$   
when  $x_i$  belongs to class 1,  $y_i = 1$

Now we have these probabilities and remember, if you were to write this hyper plane equation as the way we wrote in the last few slides  $\beta_0 + \beta_{11} X_1 + \beta_{12} X_2$ . The job of identifying a classifier as far as we are concerned is done when we identify values for the parameters  $\beta_0$ ,  $\beta_{11}$  and  $\beta_{12}$ .

So, we still have to figure out what are the values for this. Once we have a value for this, any time I get a new point I simply put it into the

$p$  of  $x$  equation that we saw in the last slide and then get a probability. So, this still needs to be identified and obviously, if we are looking at a classification problem where I have this on this side and stars on this side, I want to identify these  $\beta_0$ ,  $\beta_{11}$  and  $\beta_{12}$  in such a way this classification problem is solved. So, I need to have some objective for identifying these values. Remember in the optimization lectures I told you that that all machine learning techniques can be interpreted in some senses an optimization problem.

So, here again we come back to the same thing and then we say look we want to identify this hyper plane, but I need to have some objective function that I can use to identify these values. So, these  $\beta_0$ ,  $\beta_{11}$  and  $\beta_{12}$  will become the decision variables but I still need an objective function. And as we discussed before when we were talking about the optimization techniques, the objective function has to reflect what we want to do with this problem. So, here is an objective function looks little complicated, but I will explain this as we go along. So, I said in the optimization lectures we could look at maximizing or minimizing. In this case, what we going to say is I want to find value for  $\beta_0$ ,  $\beta_{11}$  and  $\beta_{12}$  such that this objective is maximized.

So, take a minute to look at this objective and then see why someone might want to do something like this. So, when I look at this objective function, let us say I again draw this and then let us say I have these points on one side and the other points on the other side. So, let us call this class 0 and let us call this class 1. So, what I would like to do is I want to convert this decision function into probabilities. So, the way I am going to think about this is, when I am on this line I should have the probability being = 0.5, which basically says that if I am on the line I cannot make a choice between class 1 and class 2.

Because the probability is exactly 0.5. So, I cannot say anything about it now. What I would like to do, is you can interpret it in many ways one thing would be to say, as I go away from this line in this direction, I want the probability of the data belonging to class 1 to keep decreasing. The moment that the probability that the data belongs to class 1 keeps decreasing, that automatically means since there are only 2 classes and this is the binary classification problem, the probability that the data belongs to class 0 keeps increasing.

So, if you think of this interpretation whereas, I go from here. So, here the probability that the data point belongs to class 1 let us say it is 0.5, then basically it could either belong to class 1 or class 0. And if it is such that the probability keeps decreasing here, of the data point belonging to class 1, then it has to belong to class 0. So, that is the basic idea. So, in other words we could say the probability function that we defined before should be such that whenever a data point belongs to class 0 and I put that into that probability expression, I want

a small probability. So, it might interpret the probability as the probability that the data belongs to class 1 for example, and whenever I take a data point from this side and, put it into that probability function, then I want the probability to be very high because I want that as the probability that the data belongs to class 1. So, that is the basic idea.

So, in other words we can paraphrase this and then say for any data point on this side belonging to class 0, we want to minimize  $p$  of  $x$  when  $x$  is substituted into that probability function and, for any point on this side when we substitute these data points into the probability function, we want to maximize the probability. So, if you look at this here what they say is if this data point belongs to class 0 then  $y_i$  is 0. So, whenever a data point belongs to class 0 anything to the power 0 is 1 so, this will vanish. So, in the product there will be functions of this form, which will be  $1 - p$  of  $x_i$  and because  $y_i$  is 0 this will become 1. So, this will become something to the power 0 1. So, this term will vanish and the only thing that will remain is  $1 - p$  of  $x_i$ . So, if we try to maximize  $1 - p$  of  $x_i$ , then that is equivalent to minimizing  $p$  of  $x_i$ . So, for all the points that belong to class 0 we are minimizing  $p$  of  $x_i$ .

Now, let us look at the other case of a data point belonging to class 1, in which case  $y_i$  is 1 so,  $1 - 1 = 0$ . So, this term will be something to the power 0 which will become 1. So, it cannot drop out. So, the only thing that will remain is  $p$  of  $x_i$  now  $y_i$  is 1. So, power 1 will be just left with  $p$  of  $x_i$ . And since this data belongs to class 1, I want this probability to be very large. So, when I maximize this it will be large number.

So, you have to think carefully about this equation. There are many things going on here, number 1 that this is a multiplication of the probabilities for each of the data point. So, this includes data points from class 0 and class 1. The other thing that you should remember is let us say I have a product of several numbers, if I am guaranteed that every number is positive right, then the product will be maximized when each of these individual numbers are maximized. So, that is the principle that is also operating here, that is why we do this product of all the probabilities.

However if a data point belongs to class 1, I want probability to be high. So, the individual term is just written as  $p$  of  $x_i$ . So, this is high for class 1. When a data point belongs to class 0, I still want this number to be high, that means, this number will be small. So, it automatically takes care of this as far as class 0 and class 1 are concerned. So, while this looks little complicated, this is written in this way because it is easier to write this as one expression.

Now let us take a simple example to see how this will look. Let us say I have class 0, I have 2 data points  $X_1$  and  $X_2$  and class 1, I have 2 data points  $X_3$  and  $X_4$ . So, this objective function when it is written out would look something like this. So, when we take let us say these

points belonging to class 0 then I said the only thing that will be remaining is here. So, this will be  $1 - p$  of  $X_1$  for the second data point it will be  $1 - p$  of  $X_2$ , then for the data third data point it will be  $p$  of  $X_3$  and for the fourth data point it will be  $p$  of  $X_4$ .

So, this would be the expression from here. So, now when we maximize this, then since  $p$  of  $X$ 's are bounded between 0 and 1, this is a positive number, this is a positive number, positive number positive number and, if the product has to be maximized, then each number has to be individually maximized. That means, this has to be maximized. So, it will go closer and closer to 1 the closer to 1 it is better. So, you notice that  $X_4$  would be optimized to belong in class 1 Similarly  $X_3$  would be optimized to belong in class 1 and when you come to these two numbers, you would see that this would be a large number if  $p$  of  $X_1$  is a small number. So,  $p$  of  $X_1$  basically means that  $X_1$  is optimized to be in class 0. And similarly  $X_2$  is optimized to be in class 0. So, this is an important idea that we have to understand in terms of how this objective function is generated.

(Refer Slide Time: 24:47)

Data science for Engineers

## Log-likelihood function

- The log-likelihood function will become  

$$l(\beta_0, \beta_1) = \sum_{i=1}^n y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i))$$
- Simplifying this expression and using the definition for  $p(x)$  will result in an expression with the parameters of the linear decision boundary
- Now the parameters can be estimated by maximizing the above expression using any nonlinear optimization solver

Logistic regression 10

Now, one simple trick you can do is take that objective function and take a log of that and then maximize it. So, if I am maximizing a positive number  $X$ , then that is equivalent to maximizing log of  $X$  also. So, whenever this is maximized that will also be maximized, the reason why you do this it makes the product into a sum makes it looks simple. So, remember from our optimization lectures, we said we got a maximise this objective. So, we always write this objective in terms of

decision variables and the decision variables in this case are  $\beta_0$ ,  $\beta_1$  and  $\beta_2$  so, we described before. So, what happens is each of these probability expressions, if you recall from your previous slides, will have these 3 variables and  $x_i$  are the points that are already given. So, you simply substitute them into this expression.

So this whole expression would become a function of  $\beta_0$ ,  $\beta_1$  and  $\beta_2$ . Now we have come back to our familiar optimization territory, where we have this function which is a function of these decision variables, this needs to be maximized and this is an unconstrained maximization problem because we are not putting any constraints  $\beta_0$ ,  $\beta_1$  and  $\beta_2$ . So, they can take any value that we want and also the fact that the way the probability is defined, this would also become a non-linear function. So, basically we have a non-linear optimization problem in several decision variables and, you could use any non-linear optimization technique to solve this problem and when you solve this problem, what you get is basically the hyper plane. So, in this case it is a two dimensional problem. So, we have 3 parameters. Now if there is an n dimensional problem, if you have let us say n variables.

So, I will have something like  $\beta_0 + \beta_{11} x_1 + \beta_{12} x_2$  and so on +  $\beta_{1n} x_n$ , this will be an  $n + 1$  variable problem, there are  $n + 1$  decision variables, these  $n + 1$  decision variables will be identified through this optimization solution. And for any new data point, once we put that data point into the  $p(x)$  function that sigmoidal function that we have described, then you get the probability that it belongs to class 0 or class 1.

So, this is the basic idea of logistic regression. In the next lecture, I will take very simple example with several data points to show you how this works in practice and I will also introduce notion of regularization, which would help in avoiding over fitting when we do logistic regression. I will explain what over fitting means in the next lecture also, with that you will have theoretical understanding of how logistic regression works and in a subsequent lecture doctor Hemanth Kumar would illustrate, how to use this technique in R on a case study problem.

So, that will give you the practical experience of how to use logistic regression and how to make sense out of the results that you get from using logistic regression on an example problem.

Thank you and I will see you in the next lecture.

**Data science for Engineers**  
**Prof. Ragunathan Rengasamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture- 43**  
**Logistic Regression**

(Refer Slide Time: 00:13)



We will continue our lecture on Logistic Regression that we introduced in the last lecture. And if you recall from the last lecture we modeled the probability as a sigmoidal Function.

And the sigmoidal function that we used is given here and notice that this is your hyperplane equation. And in n dimensions this quantity is a scalar, because you have X elements n elements in X and n elements in  $\beta_1$  and this becomes something like  $\beta_0 + \beta_{11} x_1 + \beta_{12} x_2$  and so on  $\beta_{1n} x_n$ .

So, this is a scalar and then we saw that if this quantity is a very large negative number, then the probability is 0 and if this quantity is a very large positive number the probability is 1. And the transition of the probability at 0.5 remember I said you have to always look at it from 1 classes viewpoint.

So, let us say if you want class 1 to have high probability and class 0 is a, row prob, low probability case, then you need to have a threshold that we described before that you could convert this into a binary output by using a threshold. So, if you were to use a threshold of

0.5, because probabilities go from 0 and 1. And then you notice that this  $p$  of  $X$  becomes 0.5 exactly when  $\beta_0 + \beta_1 X = 0$ . This is because  $p$  of  $X$  then =  $e^0$  divided by 1 + equal 0 which is equal to 1 by 2.

Also notice another interesting thing that this equation is then the equation of the hyperplane. So, if I had data like this and data like this and if I draw this line any point on this line is the probability is equal to 0.5 point. That basically says that any point on this line in this 2 d case or hyperplane, in the  $n$  dimensional case will have an equal probability of belonging to either class 0 or class 1 which makes sense from what we are trying to do. So, this model is what is called a logit model.

(Refer Slide Time: 02:52)

Data science for Engineers

**Example 1**

| $X_1$ | $X_2$ |
|-------|-------|
| 1     | 1     |
| 2     | 1     |
| 3     | 1     |
| 4     | 1     |
| 5     | 1     |
| 1     | 2     |
| 2     | 2     |
| 3     | 2     |
| 4     | 2     |
| 5     | 2     |

**Class 0**

| $X_1$ | $X_2$ |
|-------|-------|
| 6     | 3     |
| 7     | 3     |
| 8     | 3     |
| 9     | 3     |
| 10    | 3     |
| 6     | 4     |
| 7     | 4     |
| 8     | 4     |
| 9     | 4     |
| 10    | 4     |

**Class 1**

| $X_1$ | $X_2$ | Class |
|-------|-------|-------|
| 1     | 3     | ?     |
| 2     | 3     | ?     |
| 4     | 4     | ?     |
| 5     | 4     | ?     |
| 3     | 3     | ?     |
| 6     | 2     | ?     |
| 9     | 2     | ?     |
| 8     | 1     | ?     |
| 7     | 2     | ?     |
| 10    | 1     | ?     |

**Test Data**

Let us take a very simple example to understand this. So, let us assume that we are given data. So, here we have data for class 0 and data for class 1 and then clearly this is a 2 dimensional problem. So, the hyperplane is going to be a line.

So, a line will separate this. And in a typical case in these kinds of classification problems this is actually called as supervised classification problem. We call this a supervised classification problem because all of this data is labeled. So, I already know that all of this data is coming from class 0 and all of this data is coming from class 1.

So, in other words I am being supervised in terms of what I should call as class 0 and what I should call as class 1. So, in these kinds of problems typically you have this and then you are given new data which is called the test data and then the question is what class does this test data belong to. So, it is either class 0 or class 1, as far as we are concerned in this example.

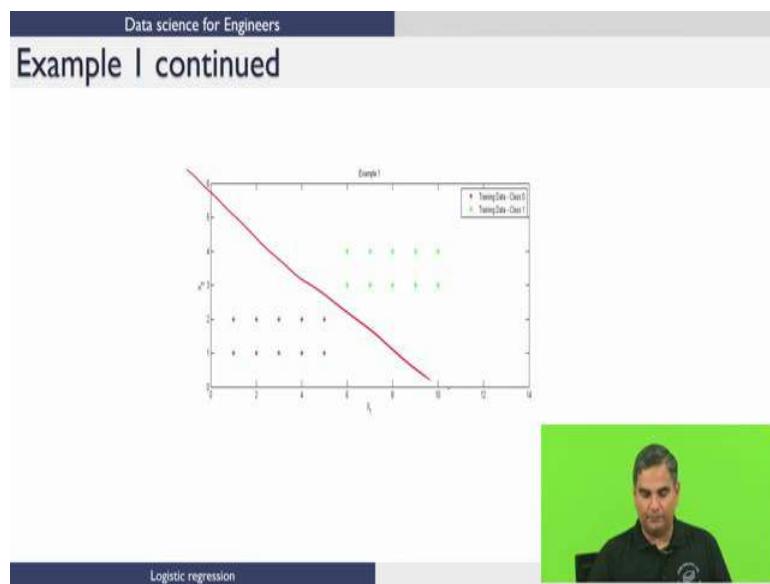
Just to keep in mind that there would we use problems like this. Remember at the beginning of this course I talked about fraud detection and so on. Where you could have lots of records of fraudulent let us say credit card use and all of those instances of fraudulent credit card use you could describe by certain attribute.

So, for example, the time of the day whether the credit card was done at, the place where the person lives credit card transfer or credit card use was done at the place the person lives and many other attributes. So, if those are the attributes let us say many attributes are there. And you have lots of records for normal use of credit card and some records for fraudulent use of credit card.

Then you could build a classifier which given a new set of attributes that is a new transaction that is being initiated, could identify what likelihood it is of this transaction being fraudulent. So that is one other way of thinking about the same problem. So, nonetheless as far as this example is concerned what we need to do is we have to fill this column with zeros and ones. If I fill a column with row with 0 then that means, this data belongs to class 0 and if I fill it 1 then let us say this belongs to class 1 and so on.

So, this is what we are trying to do, we do not know what the classes are.

(Refer Slide Time: 05:34)



So, just so let me see this it is a very simple problem we have plotted the same data that was shown in the last table. And you would notice that if you wanted a classifier, something like this would do. So, this problem is linearly separable. So, you could you could come up with a line that does it. So, let us see what happens if we use logistic regression to solve this problem.

(Refer Slide Time: 06:04)

| Test Results   |                |        |       |
|----------------|----------------|--------|-------|
| X <sub>1</sub> | X <sub>2</sub> | Prob   | Class |
| 1              | 3              | 0.0002 | 0     |
| 2              | 3              | 0.004  | 0     |
| 4              | 4              | 0.999  | 1     |
| 5              | 4              | 0.999  | 1     |
| 3              | 3              | 0.076  | 0     |
| 6              | 2              | 0.0172 | 0     |
| 9              | 2              | 0.991  | 1     |
| 8              | 1              | 0.0002 | 0     |
| 7              | 2              | 0.251  | 0     |
| 10             | 1              | 0.0667 | 0     |

So, if you did a logistics regression solution, then in this case it turns out that the parameter values are these. And how did we get these parameter values? These parameters values are guard through the optimization formulation, where 1 is maximizing log likelihood with  $\beta_0$ ,  $\beta_{11}$  and  $\beta_{12}$  as decision variables.

And as we see here there are 3 decision variables, because this was A<sub>2</sub> dimensional problem. So, 1 coefficient for each dimension and then 1 constant. Now once you have this then what you do is, you have your expression for p of X which is as written before the sigmoid. So, this is a sigmoidal function that we have been talking about. Then whenever you get a test data, let us say 1 3, you plug this into this sigmoidal function and you get a probability. Let us say the first data point when you plug in you get a probability this.

So, if you use a threshold of 0.5 then what we are going to say is anything less than 0.5 is going to belong to class 0 and anything greater than 0.5 is going to belong to class 1. So, you will notice that this is 0 class 0, class 1, class 1, class 0, class 0, class 1, class 0, class 0, class 0.

So, as I mentioned in the previous slide what we wanted was to fill this column and if you go across row then it says that particular sample belongs to which class. So, now, what we have done is we have classified these test cases, which the classifier did not see while you were identifying these parameters.

So, the process of identifying these parameters is what is usually called in machine learning algorithms as training. So, you are training the classifier to be able to solve test cases later. And the data that you use while these parameters are being identified are called the training data and this is called the test data that you are testing a classifier with.

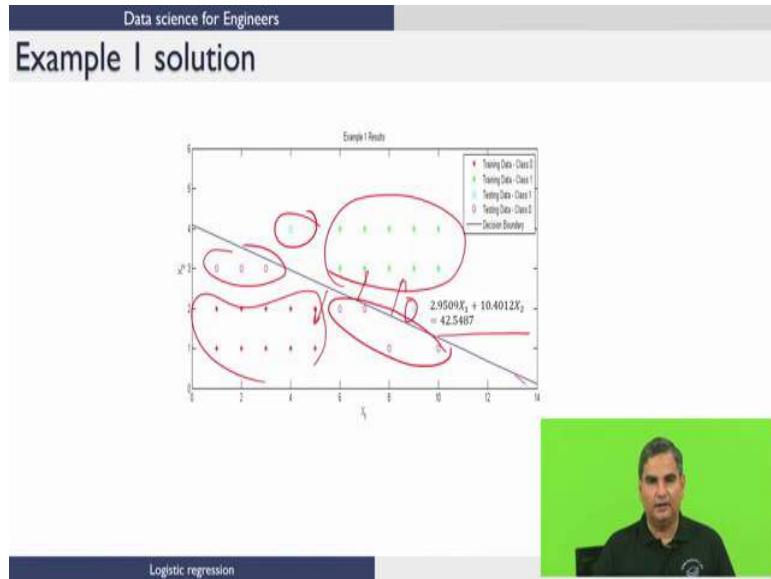
So, typically what you do is if you have lots of data with class labels already given one of the good things, you know that one should do is to split this into training data and the test data. And the reason for splitting this into training and test data is the following. In this case if you look at it, we built a classifier based on some data and then we tested it on some other data, but we have no way of knowing whether these results are right or wrong.

So, we just have to take the results as it is. So, ideally what you would like to do is, you would like to use some portion of the data to build a classifier. And then you want to retain some portion of the data for testing and the reason for retaining this is because the labels are already known in this.

So, if I just give this portion of the data to the classifier, the classifier will come up with some classification. Now that can be compared with the already established labels for those data points. So, from verifying how good your classifier is it is always a good idea to split this into training and testing data. What proportion of data you use for training, what proportion of data used for testing and so on are things to think about.

Also there are many different ways of doing this validation as one would call it with test data. There are techniques such as k fold validation and so on. So, there are many ways of splitting the data into train and test and then verifying how good your classifier is. Nonetheless the most important idea to remember is that one should always look at data and partition the data into training and testing so that you get results that are consistent.

(Refer Slide Time: 10:23)



So, if one were to draw these points again that that we use this in this exercise. So, these are all class 1 data points these are class 0 data points and this is your hyperplane that a logistic regression model figured out and these are the test points that we tried with this classifier. So, you can see that in this case everything seems to be working well, but as I said before you can look at results like this in 2 dimensions quite easily.

However, when there are multiple dimensions it is very difficult to visualize where the data point lies and so on. Nonetheless so, it gives you an idea of what logistic regression is doing. It is actually doing a linear classification here. However, based on the distance in some sense from this hyperplane. We also assign a probability for the data being in a particular class.

Now, there is one more idea that we want to talk about in logistic regression. This idea is what is called as regularization. The idea here is the following. If you notice the objective function that we used in the general logistic regression, which is what we called as a log likelihood objective function.

(Refer Slide Time: 11:42)

Data science for Engineers

## Regularization

- General objective
  - $\min_{\theta} -L(\theta)$
  - where  $L(\theta) = \left( \sum_{i=1}^n y_i \log(p(x_i)) + (1 - y_i) \log(1 - p(x_i)) \right)$
- When large number of independent variables are present, logistic regression tends to over-fit
- To prevent over-fitting, we need to penalize the coefficients
- This is known as regularization

$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$

Logistic regression 7

Here  $\theta$  again speaks to the constants in the hyperplane or the decision variables and this is the form of the equation that we saw in the previous lecture and in the beginning of this lecture also I believe. Now, if you have  $n$  variables in your problem or  $n$  features or  $n$  attributes then the number of decision variables that you are identifying are  $n + 1$ . So, 1 constant for each variable and the constant if this  $n$  becomes very large when there are large number of variables that are present then, what happens is this logistic regression models can overfit because there are so many parameters that you could tend to overfit the data.

So, to prevent this what we want to do is somehow we want to say though you have this  $n + 1$  decision variables to use, one would want these decision variables to be used sparingly. So, whenever you use a coefficient for a variable, for the classification problem, then we want ensure that you get the maximum value for using that variable in the classification problem. So, in other words if let us say there are 2 variables I say  $\beta_0 + \beta_1 x_1 + \beta_2 x_2$ . Then for this classifier, I am using both let us say variables  $x_1$  and  $x_2$  as being important. What I would like to do is make sure that I use these only if they really contribute to the solution or to the efficacy of the solution.

So, one might say that for every term that you use, you should get something in return or in other words if you use a term and get nothing in return I want to penalize this term. So, I want to penalize these coefficients. This is what is typically called as regularization.

(Refer Slide Time: 14:23)

Data science for Engineers

## Regularization

- Regularization helps in building non-complex models that avoids capturing noise in model due to over-fitting
- The objective now becomes  $\min_{\theta} -L(\theta) + \lambda * h(\theta)$  where  $\lambda$  is regularization parameter and  $h(\theta)$  is regularization function
- Depending on  $h(\theta)$ , the regularization can be classified as  $L_1$  or  $L_2$  type
- $h(\theta) = \theta^T \theta$  for  $L_2$  type regularization
- Larger the value of  $\lambda$ , more is the regularization strength
- Regularization helps the model work better on test data due to the fact that over-fitting is minimized on training data

Logistic regression

So, regularization avoids building complex models or it helps in building non-complex models. So that your over fitting effects can be reduced. So, how do we penalize this? So, notice that what we are trying to do is we are trying to minimize a log likelihood.

So, what we do here is we add another term to the objective and  $\lambda$  is called the regularization parameter and this  $h(\theta)$  is some regularization function. So, what we want to do is, when I choose the values of  $\theta$  to be very large, I want this function to be large So, that the penalty is more or whenever I choose a variable right away a penalty kicks in.

And this penalty should be offset by the improvement I have in this term of the objective function. So, that is the basic idea behind regularization. Now this function could be of many types if you use this function to be  $\theta^T \theta$ , then this is called  $L_2$  type regularization. So, in the previous example this will turn out to be  $\theta = (\beta_0 \beta_1 \beta_2)^T (\beta_0 \beta_1 \beta_2)$ .

So, in this case  $h(\theta) = \beta_0^2 + \beta_1^2 + \beta_2^2$ . Now there are other types of regularization that you can use. You can use this is what is called the  $L_2$  type or  $L_2$  norm you can also use something called an  $L$  type or  $1$  or  $L_1$  norm. And larger the value of this coefficient that is multiplying this the more is regularization strength that is you are penalizing for use of variables lot more. And one general rule is regularization helps the model work better with test data because you avoid over fitting on the train data. So, that is in general something that one can keep in mind as one does these kinds of problems.

So, with this the portion on logistic regression comes to an end what we are going to do next is we are going to show you an example case study, where logistic regression is used for a solution. However, before

we do this case study since all the case studies on classification and clustering will involve looking at the output from the r code, I am going to take a typical output from the r code and there are several results that will show up. These are called performance measures of a classifier. I am going to describe what these performance measures are and how you should interpret these performance measures once you use a particular technique for any case study.

So, in the next lecture we will talk about these performance measures and then following that will be the lecture on a case study using logistic regression.

Thank you for listening to this lecture and I will see you in the next lecture.

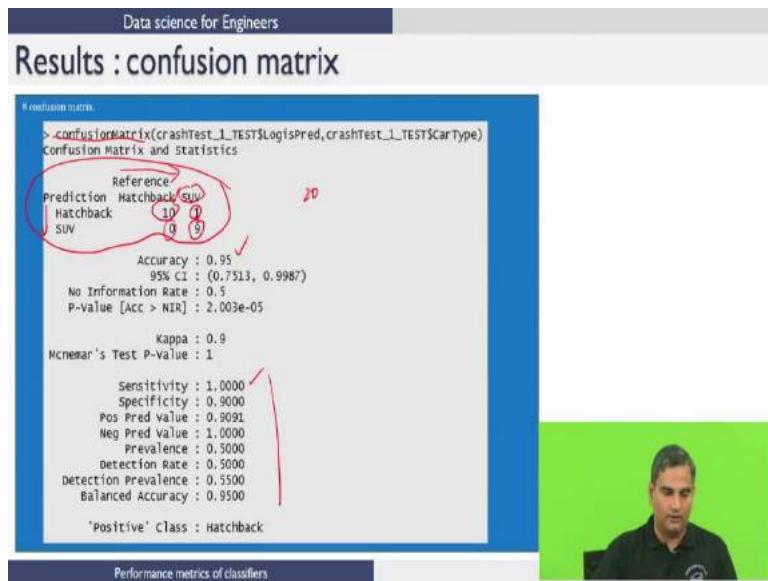
**Data science for Engineers**  
**Prof. Ragunathan Rengasamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture- 44**  
**Performance measures**

In this lecture, we will talk about typical performance measures that are used once a classifier is built. This idea is useful for all kinds of classifiers and in fact, some of these ideas could be used to benchmark different classifiers.

You will see a result like this,

(Refer Slide Time: 00:46)



when you run, r code for any of the classifiers that you are going to see as the teaching assistants describe how to do case studies and generate this table. The intention in this lecture is the following; since, you are going to see a result like this, for most classification problems. What I want to do, is I want to really, look at all of these, terms here. So, for example, there is accuracy, what does sensitivity means, specificity means positive predictive value and so on mean.

So, I am going to, first, describe what these mean in this lecture and once you understand this, in all the future lectures, whenever you look at this, you will see, how well your classifier is doing. So, let us look at, the first thing on this slide, which is something like this here. So, this is a case study which you will see later where, based on certain

attributes of a car, you are trying to classify whether it is a Hatchback or an SUV.

So, that is the kind of example problem that we have here. So, really the two classes are Hatchback and SUV. And this table that you see right here is what is called as the confusion matrix. So this is the result that you typically get. So, the way to interpret this is the following. So, if you go down this path this is what the classifier predicts for a given data and this direction is the actual label for that class.

So, for example, these is a result of one classification algorithm and let us try and see how we interpret this result. So, if you notice the total number of data points that were used in this classification algorithm can be easily found out by summing all the four elements here. So, the number of data points that were used in this algorithm are 20 and then we could interpret each one of these numbers. So, this number basically is a prediction that a car is a Hatchback and this basically says, that car was truly a Hatchback. So, these are all right predictions of Hatchbacks.

So, the prediction was Hatchback and the car was also Hatchback. Now, if you look at this number, because we are going across this row, the prediction still remains to be Hatchback. So, the classifier predicted this car to be hatchback, however, the reference is really SUV. So, this is a wrong classification of an SUV as a Hatchback. Now, if you go to the next row and then look at this, you would see that the prediction is a SUV and the reference is Hatchback. So, this 0 means there was no car which was a hatchback, which was predicted as an SUV. So, that is why you get a 0 and if you go to this number, the true car class is SUV and the prediction is also SUV. So, 9 SUVs were correctly predicted to be SUVs. So, that is what this means. Now, we are going to define some terms as we go along and we will come back to the same matrix to show you how these calculations are done and show you what the meaning of these definitions, these are so that, whenever you look at the results of another classification algorithm, you are able to kind of judge whether the algorithm did well or not and so on.

(Refer Slide Time: 05:25)



So, the previous table when we put this in this form, remember when I said reference in the previous table, this was a true condition. So, this was the true condition, this was Hatchback, this was the true condition, this was SUV and so on and this is the predicted. So, in this down, you get prediction, this is the truth. So, if you take this right here, the true condition is positive. The predicted condition is also positive. So, this is what we call as a true positive result.

So, notice, something important. This positive, the second word actually relates to the prediction and the first word refers to the truth or non truth of the prediction. So, if you say true positive, it has both information about the prediction and the actual condition also the true condition, because if I say it is positive prediction and that is true then the actual condition should have also been positive. Now, if we come to this here since, we are on the same row, the prediction still remains positive, however, the first word says it is a false prediction of positive. So, the true condition is actually negative.

So, this is a mistake of the classifier and this is a success of the classifier. Now, if we go to the second row, the second row the predictions are all negative because predicted condition negative and if you look at this right here. The prediction is negative, but this prediction is false. So, the truth is actually the condition is positive. So, this is also a mistake or a failure of the classifier. Now, if you come to this right here, since we are on the same row, the prediction is negative and we also know that this prediction is true.

So, the true condition is also negative. So, this is again a success case. So, when we think about this this way, then basically if we have

only the diagonal elements and the off-diagonal elements are zero then, we have a perfect classifier in some sense. There are no mistakes that have been made by the classifier. Now, in statistics this is called the power of the test, this is called a type one error and this is called the type two error.

(Refer Slide Time: 08:15)

Data science for Engineers

## Measures of performance

- Terminology
  - $\underline{TP}$  → true positives,  $\underline{TN}$  → true negatives,
  - $\underline{FP}$  → false positives,  $\underline{FN}$  → false negatives  

$$N = \underline{TP} + \underline{TN} + \underline{FP} + \underline{FN}$$
  - TP – Correct identification of positive labels
  - TN – Correct identification of negative labels
  - FP – Incorrect identification of positive labels
  - FN – Incorrect identification of negative labels

Performance metrics of classifiers



Now, based on those four numbers, there are many ways of measuring the performance of a classifier. So, before we do that let us make sure, that we summarize all that we said in the last slide in, in this one slide. So, we are going to use the notation TP for true positive TN for true negative FP for false positive and FN for false negative. As I mentioned before if we say TP the prediction is positive and that the true condition is also positive. So, this is correct identification of positive labels. TN then the prediction is negative that is the truth. So, correct identification of negative labels FP, the prediction is positive and it is false. So, incorrect identification of positive labels and FN is that we predict as negative, but that is an incorrect identification. We can also see that the total number of samples that we have worked with has to be =  $TP + TN + FP + FN$ , because any label, we can classify it to one of these four possible outcomes.

(Refer Slide Time: 09:33).

Data science for Engineers

## Measures of performance

- Accuracy: Overall effectiveness of a classifier
  - $A = \frac{TP+TN}{N}$  
  - Maximum value that accuracy can take is 1
  - This happens when the classifier exactly classifies two groups (i.e.,  $FP = 0$  and  $FN = 0$ )
- Remember
  - Total number of true positive labels =  $TP+FN$
- Similarly
  - Total number of true negative labels =  $TN+FP$



Performance metrics of classifiers

So, the first definition is how accurate the classifier is. So, this accuracy is very simply defined by, in all the samples how many times did the classifier get the result right. So, we had, true positive, true negative, false positive, false negative. So, I already told you that the first letter tells you the truth or the success of the classifier. So, in these four cases the true positive and true negative are the success cases and these are failure cases. So, accuracy would be true positives + true negatives divided by the total number of samples. So, this gives you how many times did I get, did it right or how many times did the classifier get it right. Now, because N is a sum of all of these and we notice from the last slide that these are the diagonal elements and I said the best classifier is one which has 0 diagonal elements. So, when N is the sum of all these four, if these both are 0 N will become  $TP + TN$ . So, accuracy = 1 or the maximum value that accuracy can take is 1. So, this is an important measure that people use, to study the performance of classifiers.

(Refer Slide Time: 11:03)

Data science for Engineers

## Measures of performance

- Sensitivity: Effectiveness of a classifier to identify positive labels
  - $S_e = \frac{TP}{TP + FN}$
- Specificity: Effectiveness of a classifier to identify negative labels
  - $S_p = \frac{TN}{FP + TN}$
- Both  $S_e$  and  $S_p$  lie between 0 and 1, 1 is an ideal value for each of them
- Balanced accuracy.
  - $BA = (sensitivity + specificity)/2$

TP ✓ TN, FP, FN



Performance metrics of classifiers

Now the other definitions, there is a definition for sensitivity, where we want to find out how effective the classifier is in identifying positive labels alone. So, in the four cases again, let us look at it true positive, true negative, false positive, false negative. So, the classifier has effectively identified a positive label only if the identification is positive and that is the truth. So, this is when the classifier identified positive labels. So, that goes into the numerator. So, we want to find the effectiveness in identifying positive label. So, what we are wanting is, of all the positive labels that were in the data, how many times did my classifier correctly identify positive labels?

So, the new denominator has to be the total number of positive labels in the data. So, this is a positive label, this is the actual condition is also true here because this is negative and true. Here the prediction is positive, but the actual, actual condition it is false. So, this is also a negative label. Here, the prediction is negative, but that is wrong. So, the actual condition is positive. So, if you want to just take the total number of positive labels in the data that will be  $TP + FN$ . So, if you divide  $TP$  divided by  $TP + FN$ , then you get what is called sensitivity. Specificity, on the other hand is the effectiveness of classifier to identify negative labels and using the same logic, you can quite easily find that the specificity will be  $TN$  by  $FE + TN$ , because this, these are the total number of negative labels, correctly identified among all the negative labels. So, true negative will be negative labels and false positive will also be negative labels, to that is the true condition of these will also be negative. So, you get this ratio to be this, you can quite easily notice that the values of  $S_e$  sensitivity and  $S_p$  specificity will both have to be between 0 and 1 and the best result is when both are 1.

So, these are two other measures that people use. And there is also another measure, which is balanced accuracy, which is sensitivity + specificity by 2, that is an average of sensitivity and specificity. We will come back to this, because these are in some sense both things that we should look at and I will tell you strategies, where, you can get sensitivity be 1 always or specificity to be 1 always, but clearly, will also tell you that those will not be the most effective classifiers for us to use.

(Refer Slide Time: 14:14)

Data science for Engineers

## Measures of performance

- Prevalence: How often does the yes condition actually occur in our sample

$$P = \frac{TP + FN}{N}$$

- Positive predictive value: Proportion of correct results in labels identified as positive

$$PPV = \frac{(sensitivity * prevalence)}{((sensitivity * prevalence) + ((1 - specificity) * (1 - prevalence)))}$$

- Negative prediction value: Proportion of correct results in labels identified as negative

$$NPV = \frac{specificity * (1 - prevalence)}{(((1 - sensitivity) * prevalence) + ((specificity) * (1 - prevalence)))}$$

Performance metrics of classifiers

Then there are other measures, there is measure called prevalence, which talks about how often does this condition actually occur in our sample. So, how many positive labels are there, totally in our sample. So, true positive is a positive label and false negative is also false positive label, because, the prediction was negative, but that is false. So, the true condition is actually positive. So, that divided by the total number in the sample space, it will give you the prevalence. Now, positive predictive value is the following. If the classifier identified several labels are as positive, what proportion of this is actually a correct result is what is. So, if all of these are identified as positive by the classifier, there is a proportion of this, which is correct. The proportion of correct results in labels identified as positive is what is called positive predictive value. Similarly, if a classifier identifies several samples as negative, the proportion of correct results within this is what is called negative prediction value.

So, this actually, is something that is used quite a bit for example, in medical community and so on. So, basically you might understand that, that this is very important right. For example,, if you go, do a test for dengue and if you get a positive result, how likely is that result to be

correct is given by, this kind of number and so on. So, these are important other measures that one could use.

(Refer Slide Time: 16:05).

Data science for Engineers

## Measures of performance

- Detection rate:  
$$DR = \frac{TP}{N}$$
- Detection prevalence: prevalence of predicted events  
$$DP = \frac{TP+FP}{N}$$
- The Kappa statistic (or value) is a metric that compares an **observed accuracy** with an **expected accuracy** (random chance)
- $$\text{Kappa} = \frac{\text{observed accuracy} - \text{expected accuracy}}{1 - \text{expected accuracy}}$$



Performance metrics of classifiers

Then there is something called a detection rate, which is defined as true positives divided by total number of samples and detection prevalence, which is true positive + false positive divided by N. So, these are easy to calculate and there are interpretations for this. The last number that we are going to talk about is what is called a K number or K statistic, which basically in some sense benchmarks whatever result you get with a random chance based classifier.

So, this is little more complicated than the other measures. So, I am just going to define this and then going to show you the formula for this. So, what you want to know is this K gives you the observed accuracy, whatever the classifier gives you as a result - what accuracy, you would expect for a classifier, which is designed based on this notion of random chance, divided by 1 - expected accuracy. So, this is the definition of K. So, what you want to have is this observed accuracy to be larger than expected accuracy.

(Refer Slide Time: 17:22).

Data science for Engineers

## Measures of performance

- Observed accuracy
  - $OA = \frac{a+d}{N}$
- Expected accuracy
  - $EA = \frac{(a+c)(a+b)+(b+d)(c+d)}{N}$
- Kappa
  - $Kappa = \frac{\frac{(a+d)}{N} - \left( \frac{(a+c)(a+b)+(b+d)(c+d)}{N} \right)}{1 - \left( \frac{(a+c)(a+b)+(b+d)(c+d)}{N} \right)}$

Where  $a, b, c$  and  $d$  are TP, FP, FN and TN respectively



Performance metrics of classifiers

So, if this is a slightly more complicated formula. So, if I have abcd defined as true positive, false positive, false negative, true negative, then the calculation for K is this. You do not have to do this calculation. This comes out of the code, but just so that you know, what these numbers are.

(Refer Slide Time: 17:50).

Data science for Engineers

## Results : confusion matrix

```
confusion matrix
> confusionMatrix(crashTest_1_TEST$LogisPred,crashTest_1_TEST$CarType)
Confusion Matrix and Statistics
Reference
Prediction Hatchback SUV
Hatchback 10 1
SUV 0 9
 TP=10 FP=1
 FN=0 TN=9
Accuracy : 0.95
95% CI : (0.7513, 0.9987)
No Information Rate : 0.5
P-value [Acc > NIR] : 2.003e-05
Kappa : 0.9
McNemar's Test P-value : 1
Sensitivity : 1.0000
Specificity : 0.9000
Pos Pred Value : 0.9091
Neg Pred Value : 1.0000
Prevalence : 0.5000
Detection Prevalence : 0.5500
Balanced Accuracy : 0.9500
'positive' Class : Hatchback
```



Performance metrics of classifiers

So, let us go back to the same example, that we had and then look at, these numbers for, for this example, this example; we talked about hatchback and SUV. Clearly, we are not talking about a positive label, a negative label here. However, if you want to use, these measures, you have to make one of these a positive label and the one, a negative label, you could make either one positive or negative, but whenever, there is a result like this. That is shown in, in R, then the, the first one is the positive label and the second one is the negative label. In fact, you can see that here positive class = hatchback. So, this is the positive label.

So, now let us look at this and then see whether we can do all these calculations for this example. So, let us look at this number. We will see, what this is. So, the prediction is hatchback and the truth is also hatchback. So, this is true positive. Now, here the prediction is hatchback, but the truth is SUV. So, this is a false positive and here the prediction is a SUV and the truth is hatchback. So, this is what I would call as false negative and here the prediction is SUV and the truth is also SUV. So, we will call this as the true negative. So, true positive = 10, false positive = 1, false negative = 0 ,true negative = 9. So, this is what we have this now, let us go through the formulae that we had before and then see whether all of this fits in. So, the accuracy is the number of times we got it right. So, in this case, if I sum up all the diagonal elements, which will be true positive + true negatives.

Those are the number of times, we got this right. So, the numerator for accuracy will be 19 divided by the total number of samples, which is 20. So, you see that 0.95 answer for this. Now, when we look at sensitivity, we said sensitivity is defined as true positive divided by true positive + false negative. So, this is how we define sensitivity. So, this is going to be = 10 divided by 10 + 0. So, you get 1 here. Similarly, you can verify the specificity to be 0.9. Now, let us look at the positive predictive value. So, the positive predictive value is one where, of all the labels that were identified as positive. How many of them were actually true positive. So, that is what this would be.

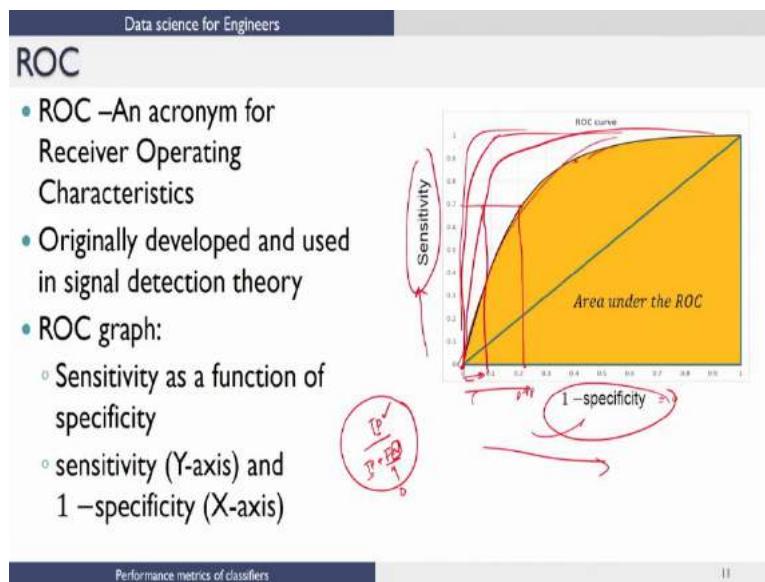
So, this would be true positive divided by true positive + false positive. Because we are talking about all the labels that are identified as positive that is, this number of which how many are true. So, if you look at this then, we will substitute this here. So, the true positive is 10 and false positive is 1. So, 10 divided by 11 and you will get this number 0.9091. And the negative predictive value would be the number of times that the negative prediction is right. Among all the predictions for as negative. Very similar to the positive predictive value and you can use the formula that I shown before to use this. In fact, this negative predictive value will be true negative, by true negative + false negative.

So, again of all the labels that are predicted as negative, how many are actually correct? So, if you do this calculation. So, true negative is

9 divided by true negative, 9 + false negative 0. So, 9 by 9, which is 1, which is what you see here. The other ones prevalence detection rate, detection prevalence and, balanced accuracy are very-very simple calculations based on the formulae, that we have in the slide, of this lecture. So, this kind of, gives you an idea of how to interpret the results that come out of, a R code, for a classifier. Now, also you notice that, this K value is here and based on the complicated formula that I showed you before.

Now, if one were to ask a question as to, what are good values for this, then that I, where a little bit of subjectivity comes in. There are applications, where, you might say, sensitivity is very important or there might be applications very much, say specificity is little more important than sensitivity and so on. So, it is kind of application dependent and depending on what you are going to use these results for, that is something that you should really think about, before finding out which, which of these numbers are important. From your application viewpoint, nonetheless, what we wanted to do was in one lecture kind of, give you the calculations for all of these. So, that, it is a handy reference for you, when you work with the case studies, in R. One last curve, which is seen in, in many papers and reported is this curve called ROC, which is an acronym for receiver operating characteristics and this was originally developed and used in, signal detection theory.

(Refer Slide Time: 24:01).



What this ROC curve is, is, a graph between sensitivity and  $1 - \text{specificity}$ . So, it is important to notice that this is  $1 - \text{specificity}$ . So clearly, we know that the best value for sensitivity is 1 and the best value for specificity is also 1. So, ideally you want both of these to be

one that would mean that you know, as you go, along this sensitivity curve.

So, for example, if you take a particular sensitivity, this is the ROC curve, this right here, this is ROC curve. So, if you take a particular sensitivity. So, you push your sensitivity to be more and more let us say, you go to 0.7 then the, specificity is, let us say this is 0.22, something like that. So, let us say, the specificity is 0.78, because 0.22 is 1 - specificity. So, the way to think about this is the, the best specificity point, is actually this right, because 0 1 - specificity is 0 would tell me specificity is 1. So, this is best point for specificity, but it is the worst point for sensitivity, because sensitivity is 0. Now, as you try to push your sensitivity to be more and more, then if you are sitting on this curve, you are going away from the best specificity point.

So, this happens to be the best specificity worst sensitivity and as you push this sensitivity more and more, you go further away from your best specificity point. So intuitively, if you want a good ROC curve, then what you want is the slope here to be, something like this. So, as I go away from sensitivity, I do not want to lose too much specificity. So, if the curve becomes something like this, it is better, because at the same 0.7, if you notice, I have given up only this much. Whereas, for this curve, I have to give up this much and if it is even sharper, then you give up less and less.

So, this curve kind of benchmarks different classifiers so, that is the most important thing to remember. Another thing to remember is, if you told me that I want the best sensitivity, I do not care about specificity, then that is a very trivial solution. The reason is, remember, how sensitivity is defined? Sensitivity is defined as TP by TP + false negative. So, this is how sensitivity is defined. So, how many times do I get true positive divided by true positive by + false negative. Now, think about this, if I want to make this one, which is the best number for sensitivity, then my strategy is very simple, I will do no classification. Every label, I will simply call it positive. Now, if I do that then let us see what happens to this. Notice an important thing, I said this is what the classifier predicts and this is the truth or the falsity of what the prediction is. Now, if I come up with a strategy, a classifier, which simply says positive for every label, let us see what happens to sensitivity. So, you will have true positive on the numerator divided by true positive, and this false negative will be 0 and the false negative will be 0, because negative speaks to the prediction, but I have a classifier, where I am never going to play it like negative.

So, this will be 0. So, it will be true positive by true positive sensitivity will be 1. Similarly, if I come up with a classifier, which does nothing, but says everything is negative label without doing anything, then that classifier will have specificity value of 1 right. So, if I want to get a sensitivity value of 1, I simply come up with a classifier, which does nothing but says everything is a positive label

and if I want, a specificity of 1, I come up with a classifier which says every label is negative without doing anything. So, both of these classifiers are useless.

So, there has to be some give take and that given take is what is shown by this curve right here and if you take a normalized area and then say this area under ROC is this yellow portion, then you would notice that better and better ROC curves are things like this. So that means, as the area under the Roc curve goes closer and closer to 1, I am getting better and better classifier designs.

(Refer Slide Time: 29:47).

Data science for Engineers

## ROC

- ROC can be used to
  - See the classifier performance at different threshold levels (from 0 to 1)
  - AUC- Area under the ROC
    - An area of 1 represents a perfect test; an area of 0.5 represents a worthless model.
    - .90 – 1 = excellent ✓
    - .80 – .90 = good ✓✓
    - .70 – .80 = fair ✓✓
    - .60 – .70 = poor ✓✓
  - AUC < 0.5, check whether your labels are marked in opposite

ROC curve

Sensitivity

1 -specificity

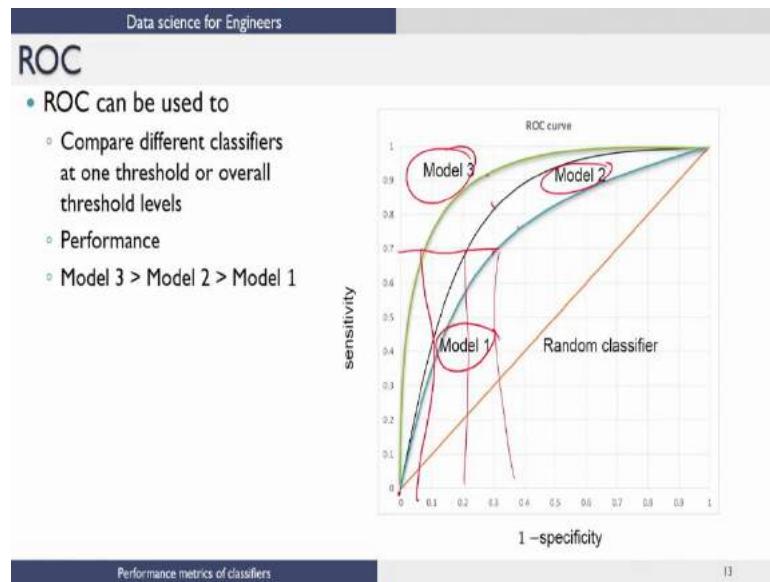
Area under the ROC

Performance metrics of classifiers

12

So, in general, if the area under the curve is between 0.9 to 1, we would call that an excellent classifier and similarly, definitions for good, fair and poor. If actually the ROC curve goes below this line, then there is some serious problem. So, you might want to go and check whether there is anything wrong with your data and so on.

(Refer Slide Time: 30:20).



So, this is for a single classifier, but if you have several classifiers that you are using, then I would say this classifier model one. This is classifier with model 2 classifier, model 3, then this classifier is better than this Classifier is better than this classifier, because if you take at any sensitivity level, if I go across the amount I give up in specificity, because this is the best specificity point for classifier 3 is less than this, is less than this. So, if I have to pick this to get the same sensitivity, I have to give a lot more of specificity. So, that is a key idea, when you try to benchmark different classifiers in terms of their performance.

So, I hope, this gives you an idea of, how you can benchmark the performance of various classifiers and how to interpret numbers that one would typically see with, with the confusion matrix and so on. So, this is an important lecture for you to understand so that, when these case studies are done and when the results are being presented, you will know, how to interpret them and understand these results, thank you very much.

In the next lecture, after an case study on logistics regression is presented to you, I come back and talk about two different types of techniques. One is called K means clustering, the other one is really just looking at neighborhood and doing classification in a very nonparametric fashion, which is called the K nearest neighbor approach. So, I will talk about both of these, in later lectures.

Thank you.

**Data science for Engineers**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 45**  
**Logisitic Regression implementation in R**

(Refer Slide Time: 00:20)

The screenshot shows a presentation slide with a dark blue header bar at the top containing the text 'Data Science for Engineers'. Below this is a light grey header bar with the title 'In this lecture'. The main content area is white and contains a bulleted list of topics:

- Case study
  - Problem statement
- Solve the case study using R
  - Read the data from a ".csv" file
  - Understand the data
  - `glm()` function
  - Interpret the results

At the bottom of the slide, there is a dark blue footer bar with the text 'Logistic Regression in R' on the left and the number '2' on the right.

Welcome to the lecture of Implementation of Logistic Regression in R. In this lecture we are going to look at a case study and a problem statement associated with it. We are also going to solve the case study using R and as a part of this we are going to look at how to read a data from a csv le, how to understand the data and how to interpret the results.

(Refer Slide Time: 00:38)

Data Science for Engineers

## Key points from previous lecture

- Logistic Regression is primarily used as a classification algorithm
- It is supervised learning algorithm
  - Data is labelled
- Parametric approach
- Decision boundary derived based on probability interpretation
- Decision boundary can be linear/ non-linear
- Probabilities are modelled as sigmoidal function



Logistic Regression in R

Some key points from previous lecture of Professor Raghu's. We know that logistic regression is a classification technique and it is a supervised learning algorithm. By supervised I mean that the data is labelled. It is also a parametric approach since at the end of the application of the algorithm we are going to get parameters out of it.

The decision boundary is derived based on the probability interpretation and it can be linear or non-linear. The probabilities are also modeled as a sigmoidal function.

(Refer Slide Time: 01:14)

### Automotive Crash Testing



Logistic Regression in R

So, let us look at the problem statement. We are going to use automotive crash testing case to illustrate this concept.

(Refer Slide Time: 01:22)

Data Science for Engineers

# Automotive Crash Testing- Problem Statement

- A crash test is a form of destructive testing that is performed in order to ensure high safety standards for various cars

A medium shot of a woman with long dark hair, wearing a purple top, sitting in front of a green screen. She is looking slightly down and to her left, possibly at a script or notes. The background is a solid green color.

Logistic Regression in R

So, a crash test is a form of destructive testing that is performed in order to ensure high safety standard for various cars.

(Refer Slide Time: 01:30)



Now, this is how a crash test is performed.

(Refer Slide Time: 01:34)

Data Science for Engineers

## Automotive Crash Testing- Problem Statement

- Several cars have rolled into an independent audit unit for crash test
- They are being evaluated on a defined scale {poor (-10) to excellent(10)} on:
  - 1) Manikin head impact
  - 2) Manikin body impact
  - 3) Interior impact
  - 4) HVAC impact
  - 5) Safety alarm system

Logistic Regression in R



So, several cars have rolled into an independent audit unit for crash test and they have been evaluated on a defined scale from poor to excellent with poor being - 10 and excellent being + 10. So, from - 10 to + 10 is the scale and they are being evaluated on a few parameters. So, let us look at the parameters they have been evaluated on.

So, I have the manikin head impact which is at what impact the head of the car crashes, the manikin body impact, the impact on the body of the car, the interior impact, the heat ventilation air conditioning impact and the safety alarm system.

(Refer Slide Time: 02:20)

- Each crash test is very expensive
- The crash test was performed for only 100 cars
- Type of car- Hatchback/SUV, was noted
- However with this data in future they should be able to predict the type of the car
- Part of data reserved for building a model and remaining kept for analysis



Now, each crash test is very expensive to perform and hence the company does a crash test for only 100 cars. At the end of the crash

test, the type of the car is noted. So, that type here is either hatchback or SUV. However, since the crash test is very expensive to perform every time, so the company is going to take this data build a model and with this model it should be able to predict the type of the car in future. So, for this we are going to reserve a part of the data for building a model and for training and the rest of the data will be kept for analysis.

(Refer Slide Time: 03:04)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data Science for Engineers'. Below the header is a light gray title bar with the text 'Automotive Crash Testing'. The main content area contains a bulleted list of instructions:

- Data for 80 cars is given in crashTest\_1.csv
- Data for remaining 20 cars is given in crashTest\_1\_TEST.csv
- Use [logistic regression](#) classification technique to classify the car types as Hatchback/SUV

At the bottom of the slide, there is a dark blue footer bar with the text 'Logistic Regression in R' and some small icons. To the right of the slide, there is a small video thumbnail showing a woman with long dark hair, wearing a purple top, sitting in front of a green screen.

So, for this we have 100 cars in total, out of which 80 cars is going to be taken as train and the remaining 20 cars is going to be taken as test. So, the 80 cars is given in crash test underscore 1 dot csv file and the remaining 20 cars is given in crash test underscore 1 underscore test dot csv.

Now, we need to use logistic regression technique to classify the car type as hatchback or SUV.

(Refer Slide Time: 03:37)

Data Science for Engineers

## Getting things ready

- Setting working directory, clearing variables in the workspace
- Installing or loading required packages

```
Set the working directory as the directory which
#contains the data files
setwd("Path of the directory with data files")
rm(list=ls()) # to clear the environment
install.packages("caret",dependencies = TRUE)

library(caret) # for confusionMatrix
```



Logistic Regression in R

Now, let us look into the solution approach. So, before we jump into modelling, let us get the things ready. We need to set the working directory, clear the variables in the workspace, we also need to load the required packages. So, in this case `glm` is an inbuilt function so we do not need any specific package to be loaded whereas to use the confusion matrix I need a package called `carrot` which I am going to load before I begin my modeling. I am also going to clear all the variables in the environment using this function which we have already learnt.

(Refer Slide Time: 04:18)

Data Science for Engineers

## Reading the data

- Data for this case study is provided to you in files with names
  - `crashTest_1`- training data
  - `crashTest_1_TEST`- testing data
- To read the data from a “.csv” file we use `read.csv()` function



Logistic Regression in R

So, now let us read the data. So, the data for this case study like I said is provided to you with these two file names. So, crash test underscore 1 is the train data and crash test underscore 1 underscore TEST is the test data.

(Refer Slide Time: 04:38)

Data Science for Engineers

## read.csv()

Reads a file in table format and creates a data frame from it

SYNTAX

```
read.csv(file, row.names=1)
```

|           |                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| file      | the name of the file which the data are to be read from. Each row of the table appears as one line of the file.                                                                                                                            |
| row.names | a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names. |

Logistic Regression in R

Now, to read the data from a CSV file we are going to use the `csv` function. Like I said from my earlier lecture it reads a file in table format and creates a data frame from it. So, its syntax is given below the inputs are file and column name.

(Refer Slide Time: 04:51)

Data Science for Engineers

## Reading the data

```
#Reading the data
crashTest_1<-read.csv("crashTest_1.csv",row.names=1)
crashTest_1_TEST<-read.csv("crashTest_1_TEST.csv",row.names=1)
```

Environment History Connections

Import Dataset

Global Environment

Data

- crashTest\_1 80 obs. of 6 variables
- crashTest\_1\_TEST 20 obs. of 6 variables

Logistic Regression in R

So, now let us read the data. So, I am going to use a function read dot csv followed by the name of my file. Now once this command is run, it's going to save it in an object called crash test underscore 1 which is a data frame. Similarly I do it for the other data set as well and now I have another object crash test underscore one underscore test. Now both these data frames will be reflected in the environment.

(Refer Slide Time: 05:21)

|   | ManHi | ManBl | IntI | HVAC  | Safety | CrtType   |
|---|-------|-------|------|-------|--------|-----------|
| 1 | -5.27 | -1.30 | 2.86 | -4.85 | 4.04   | SUV       |
| 2 | -4.82 | -5.38 | 9.72 | -0.97 | -4.57  | Hatchback |

Now, let us view the data. So, I am going to use the view command followed by the name of my data frame. So, this is how it appears. Once you run the command a separate tab will appear with all the variables and the values.

(Refer Slide Time: 05:40)

- crashTest\_1 contains 80 observations of 6 variables
- crashTest\_1\_TEST contains 20 observations of 6 variables
- The variables are: Manikin head impact, Manikin body impact, Interior impact, HVAC impact, Safety alarm system
  - First five columns are the details about the car and last column is the label which says whether the cartype Hatchback/ SUV

Now, let us try to understand the data. The data set crash test underscore one contains 80 observations of 6 variables and similarly crash test underscore 1 underscore TEST contains 20 observations of 6 variables. Now, like I said earlier we have 5 variables here which have been measured at the end of a crash test and if you can see from the data, the first five columns are the details about the car and the last column is the label which says whether the card type is hatchback or SUV.

(Refer Slide Time: 06:15)

Data Science for Engineers

## Structure of the data

- Structure of data
  - Variables and their data types
- `str()`

SYNTAX

`str(object)`

object      any R object about which you want to have some information.



Logistic Regression in R

So, let us look at the structure of the data. By structure I mean the variables and their corresponding data types. So, structure is the command which is represented as str. I need to give an object to it as input the object here is the desired object for which we want to look at the structure.

(Refer Slide Time: 06:36)

Data Science for Engineers

## Structure of train data

```
> str(crashTest_1)
'data.frame': 80 obs. of 6 variables:
 $ ManHI : num -5.27 -4.82 9.57 2.84 0 0.4 5.94 5.78 0.86 7.36 ...
 $ ManBI : num -1.3 -5.38 -7.5 -2.85 2.68 6.34 3.14 -1.75 -4.32 7.42 ...
 $ IntI : num 2.86 9.72 -7.61 0.92 -4.15 0.83 -6.65 -6.85 8.1 0.27 ...
 $ HVACI : num -4.85 -0.97 1.33 5.51 0.85 5.03 6.62 0.73 -8.96 -8.62 ...
 $ Safety : num 4.04 -4.57 -5.1 -6.64 5.58 -8.1 -1.32 5.5 3.1 3.08 ...
 $ CarType: Factor w/ 2 levels "Hatchback","SUV": 2 1 1 1 2 2 1 1 1 2 ...
```



Logistic Regression in R

So, if you look at the structure of the train data it tells you that crashTest\_1 is of the type data frame with 80 observations and 6 variables and all the five variables are numeric and the class variable which is carType is a factor with levels hatchback and SUV.

(Refer Slide Time: 06:59)

Data Science for Engineers

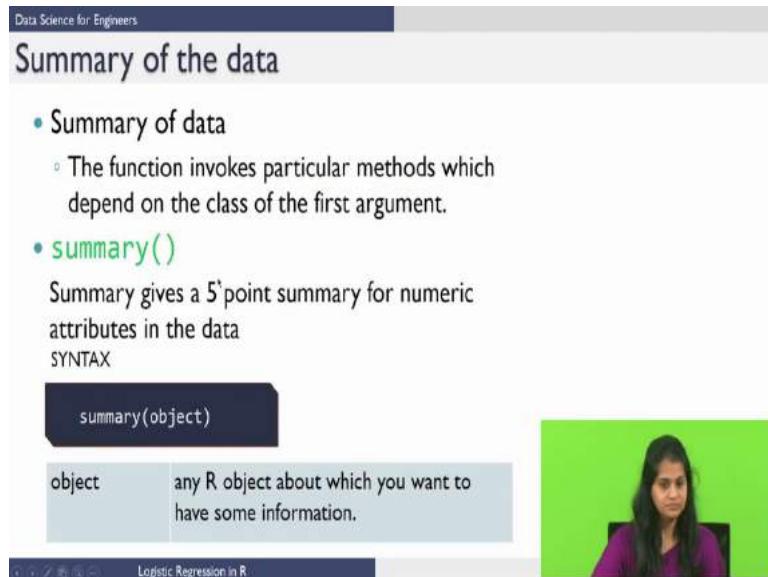
## Summary of the data

- Summary of data
  - The function invokes particular methods which depend on the class of the first argument.
- **summary()**
  - Summary gives a 5-point summary for numeric attributes in the data

SYNTAX

```
summary(object)
```

object      any R object about which you want to have some information.



Logistic Regression in R

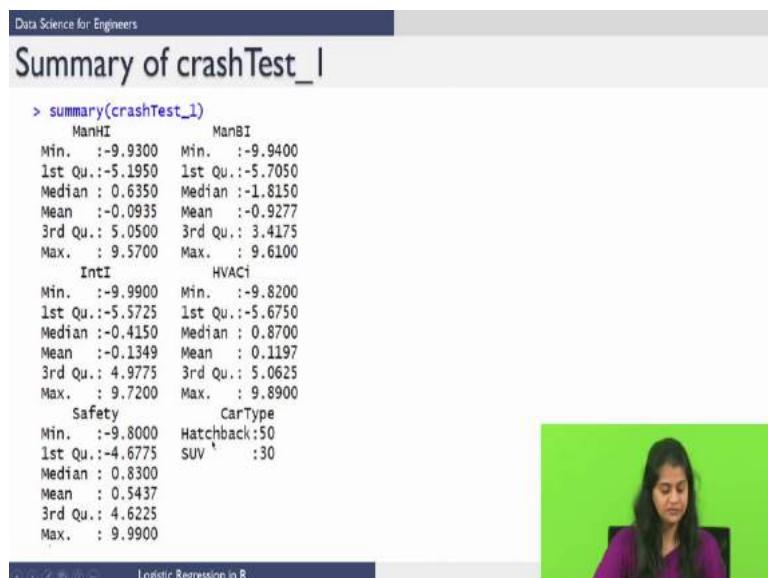
Similarly you can also look at the structure of the test set. So now, let us look at the summary of the data and let us see what it has to tell about the data. So, summary is the five point summary if the input is a data frame and if the input is an object than the corresponding summary for the object is returned. So, this is the syntax.

(Refer Slide Time: 07:20)

Data Science for Engineers

## Summary of crashTest\_1

```
> summary(crashTest_1)
 ManHt ManBt
Min. :-9.9300 Min. :-9.9400
1st Qu.:-5.1950 1st Qu.:-5.7050
Median : 0.6350 Median :-1.8150
Mean :-0.0935 Mean :-0.9277
3rd Qu.: 5.0500 3rd Qu.: 3.4175
Max. : 9.5700 Max. : 9.6100
 IntI HVACI
Min. :-9.9900 Min. :-9.8200
1st Qu.:-5.5725 1st Qu.:-5.6750
Median :-0.4150 Median : 0.8700
Mean :-0.1349 Mean : 0.1197
3rd Qu.: 4.9775 3rd Qu.: 5.0625
Max. : 9.7200 Max. : 9.8900
 Safety CarType
Min. :-9.8000 Hatchback:50
1st Qu.:-4.6775 SUV :30
Median : 0.8300
Mean : 0.5437
3rd Qu.: 4.6225
Max. : 9.9900
```



Logistic Regression in R

So, the summary for the train data which is crashTest\_1 is given below in the snippet. So, for the numerical variables it is a five point summary with minimum first quartile and median, mean, third quartile and maximum. For the categorical variable which is the factor car type here it returns the frequency count.

(Refer Slide Time: 07:43)

Data Science for Engineers

## Summary of crashTest\_1\_TEST

```
> summary(crashTest_1_TEST)
 ManHI ManBI
Min. :-9.940 Min. :-8.740
1st Qu.:-5.535 1st Qu.:-2.502
Median : 0.740 Median : 0.670
Mean : 0.047 Mean : 0.328
3rd Qu.: 5.110 3rd Qu.: 2.500
Max. : 9.090 Max. : 8.420
 IntI HVACi
Min. :-8.950 Min. :-9.2300
1st Qu.:-3.272 1st Qu.:-2.4550
Median : 1.200 Median : 0.6750
Mean : 0.524 Mean : 0.7235
3rd Qu.: 3.908 3rd Qu.: 5.3375
Max. : 8.870 Max. : 8.3300
 Safety CarType
Min. :-8.660 Hatchback:10
1st Qu.:-6.095 SUV :10
Median :-0.770
Mean : 0.191
3rd Qu.: 4.992
Max. : 9.620
```



Logistic Regression in R

So, for the test it again returns a five point summary for the numerical variables and for the car type it tells me that there are 10 cars of type hatchback and 10 cars of the type SUV.

(Refer Slide Time: 07:58)

Data Science for Engineers

## glm()

```
glm(formula, data, family)
```

**Arguments**

|         |                                                                                                                                                                                                                                                                                                                 |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| formula | object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted                                                                                                                                                                                          |
| data    | dataframe containing variables                                                                                                                                                                                                                                                                                  |
| family  | a description of the error distribution and link function to be used in the model. For <code>glm</code> this can be a character string naming a family function, a family function or the result of a call to a family function. In specific, <code>family="binomial"</code> corresponds to logistic regression |



Logistic Regression in R

So, now let us look at the function `glm` which we are going to use for logistic regression. So, `glm` stands for generalized linear model and the input to it is a formula, a data and family. So, formula is basically a symbolic representation of the model you want to fit. So, in our case it becomes the car type. So, it is basically a class. Data is a data frame from which you want to obtain your variables and family is binomial if you use logistic regression. There are also other families which are listed inside the function but if you write `family = binomial` then it specifically corresponds to logistic regression.

(Refer Slide Time: 08:45)

```
Model
logisfit<-glm(formula = crashTest_1$carType~., family = 'binomial',
 data = crashTest_1)

p(X) =
$$\frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

> logisfit
Call: glm(formula = crashTest_1$carType ~ ., family = "binomial", data = crashTest_1)

Coefficients:
(Intercept) ManHI ManBI IntI HVACi Safety
 -22.76 -13.48 36.02 -44.90 -58.50 -27.36

Degrees of Freedom: 79 Total (i.e. Null); 74 Residual
Null Deviance: 105.9
Residual Deviance: 5.359e-08 AIC: 12
```

Now, let us build a logistic regression model. Now, I am going to use the `glm` function the formula here says that get the variable car type which is our class here from the data `crashTest_1` and to access it I use a dollar symbol. Now, `crashTest_1` is my train data. Now, like I said earlier `family = binomial` corresponds to logistic regression and now the variable car type is to be obtained from `crashTest_1`.

Now, once I run this command an object of the type `glm` is created and I call it `logisfit`. So, if you could recall from Professor Raghu's lecture, we model the probabilities as a sigmoidal function and on the right hand side I have the log odds ratio and this = the hyperplane equation. This is also the decision boundary. Here  $p(x)/(1 - p(x))$  is the odds, where  $p(x)$  is the probability of success in  $(1 - p(x))$  is the probability of failure. Now,  $p(x)$  in our case is the probability that the car type is hatchback and  $(1 - p(x))$  is the probability that the car type is SUV.

Now, let us look at the model. Now if you run the model `logisfit` in your console. This is what is displayed. In the first line it displays the formula, in the next line it displays the coefficient then I have degrees

of freedom. So, it displays two degrees of freedom. So, the first degrees of freedom is when you have a null model that is only with the intercept and in the second case you have a degrees of freedom = 74 which means that I have included all the variables into my modeling.

(Refer Slide Time: 10:42)

```

Data Science for Engineers
Summary of model
> summary(logisfit)

Call:
glm(formula = crashTest_1$carType ~ ., family = "binomial", data = crashTest_1)

Deviance Residuals:
 Min 1Q Median 3Q Max
-1.316e-04 -2.100e-08 -2.100e-08 2.100e-08 1.266e-04

Coefficients:
 Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.76 12007.54 -0.002 0.998
ManHI -13.48 3077.29 -0.004 0.997
ManBI 36.02 7221.18 0.005 0.996
IntI -44.90 8853.08 -0.005 0.996
HVACi -58.50 11461.92 -0.005 0.996
Safety -27.36 5396.42 -0.005 0.996

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1.0585e+02 on 79 degrees of freedom
Residual deviance: 5.3590e-08 on 74 degrees of freedom
AIC: 12

Number of Fisher Scoring iterations: 25

```

$\widehat{\beta}_i$   
 $i = 0, 1, \dots, 5$

So, now let us look at each of the coefficients in detail. So, I am going to again use summary of logisfit. So now, this is similar to what we have done in linear regression. The first section tells you the formula that we have used, second section tells you the measure of a t and the 5 point summary for it. I have the next section as the coefficients. So, these are the  $\beta_i$ 's, where  $i = 0$  to  $5$ . Now, for the intercept it is  $\beta_0$  so on and so forth. I have 4 columns here I have the estimate, standard error, the z value and the associated probability. Now in logistic regression the coefficients gives you the change in log odds of the output for a unit increase in the predictor value which is the input value. So, now if you can see the probability is really really high and none of the variables are statistically significant.

If you go to next section I have something called null deviance and residual deviance. So, null deviance is the deviance of your model when only the interceptor is present and residual deviance is a deviance of your model when all the terms are taken into account. So, you can look at the degrees of freedom and tell whether it is a null, model that is a reduced model, or the full model.

So, for the reduced model I take only the interceptor. So, my degrees of freedom reduced by 1, so  $80 - 1$ , 79. Whereas, for the full model I take all the variables into account. So, I have  $80 - 6$  degrees of freedom which is 74. So, I have something called the Fisher scoring iteration. So, the Fisher's scoring is used for maximum likelihood

estimation and it is a derivative of Newton Raphson method. So, it tells you that the number of iterations that it has taken is 25.

(Refer Slide Time: 12:46)

Data Science for Engineers

## Finding the odds

- `predict()`
- Syntax: `predict(object)`

```
Finding the odds
logisTrain<-predict(logisfit, type = 'response')
```

- `predict()` with  
`type='response'` gives probabilities
- By default otherwise it returns log(odds)



Logistic Regression in R

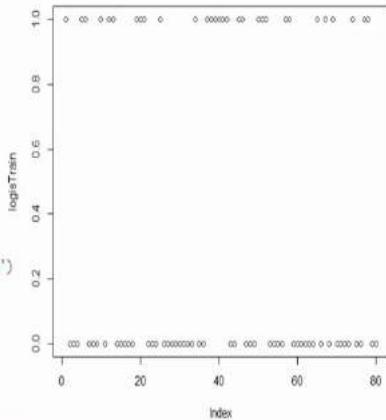
Now, let us find the odds. To find the odds we are going to use the predict function and the syntax is predict and my input is an object for which I want to predict it. Now, for our data I am going to use predict. My input here is the logistic regression model, now if I do not give any data then the function assumes that I want to predict it on the train set which is crashTest\_1 in our case. Now, type = response gives you the output as probabilities, but if you do not mention this it by default gives you the log odds.

(Refer Slide Time: 13:33)

Data Science for Engineers

## Plotting the probabilities

```
plot(logisTrain)
```



```
Finding the odds
logisTrain<-predict(logisfit, type = 'response')
```

Logistic Regression in R

Now, let us plot the probabilities. So after you run the command I have saved it as logistrain and I am going to use the plot function to plot the probabilities. On to my right I have the plot on the y axis I have the probabilities and on the x axis I have the index. So, from this plot it is clear that the classes are well separated, but we still do not know which site belongs to which car type. So, let us see how to find out which side belongs to which car type.

(Refer Slide Time: 14:09)

Data Science for Engineers

## Identifying probabilities associated with the CarType

- Mean of probabilities
- This helps us identify the probabilities associated with the two classes

```
> tapply(logisTrain,crashTest_1$CarType,mean)
 Hatchback SUV
2.851316e-10 1.000000e+00
```

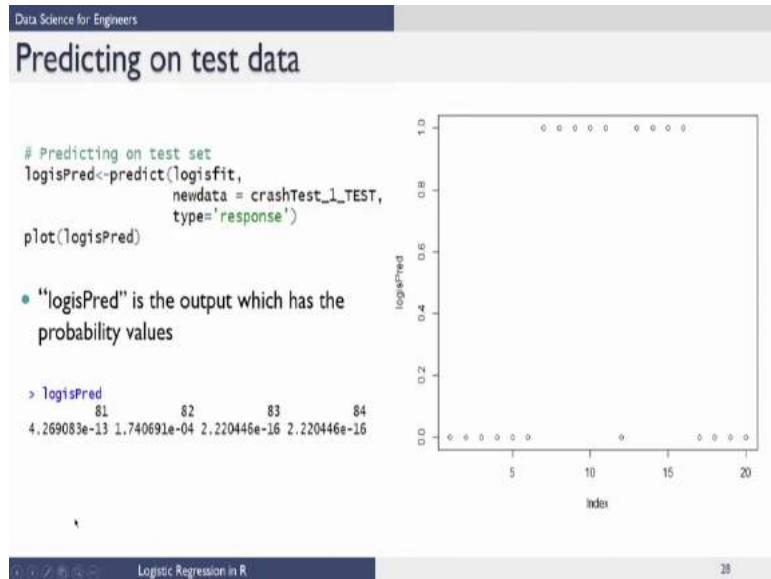


Logistic Regression in R

So, to do that I am going to find the mean of the probabilities and this will help us to identify the probabilities which are associated with the two classes. I am going to use the t apply function now this should ring a bell we will on this in the introduction to basics of R programming lecture. Then put here I give as logistrain. Now I am want to classified based on the car type. So, I am going to give crashTest\_1\$car type and the function I want to find is the mean. So, I am giving mean as the function.

Now, if you run the command I have the probabilities associated with each car type. For hatchback it is really really low its 2.85 into 10 power - 10 where as for SUV it is 1. So, this tells us that the lower probabilities are associated with the car type hatchback where as the higher probabilities are associated with the car type SUV.

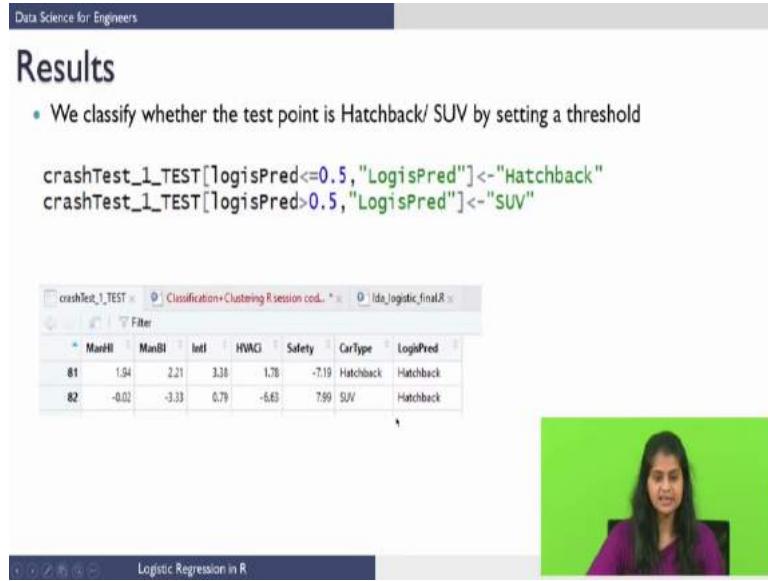
(Refer Slide Time: 15:13)



So, now let us predict this on the test data. I am again going to use the predict function. My input again is the logistic regression model. Now my new data is the test data. So, crashTest\_1\_TEST is the test set. Now, since I want to again model the probabilities, I am going to give type = response, now once this command is executed it gets stored as an object logispred and now I will plot the logispred. So, if I plot it, I have the plot on my right, again I have the predicted values of probability on my y axis and I can see that on the x I have the index.

Now, even for the test set the classes are well separated, now I know that the points which fall here belong to the class of hatchback and the points which are above belong to the class SUV. Now, logispred is the output it has the probability values and I have a small snippet below that shows me the probability values. Now, I have shown you only for the first four points you will also get similar values for the remaining points.

(Refer Slide Time: 16:28)



The screenshot shows an RStudio interface with three tabs: 'crashTest\_1\_TEST', 'Classification+Clustering R session code...', and 'ida/logistic\_final.R'. The 'ida/logistic\_final.R' tab contains the following R code:

```
crashTest_1_TEST$logisPred<=0.5, "LogisPred"]<- "Hatchback"
crashTest_1_TEST$logisPred>0.5, "LogisPred"]<- "SUV"
```

Below the code, a data frame is displayed with columns: MarshI, ManBl, InfI, HWAG, Safety, CarType, and LogisPred. Two rows are shown:

|    | MarshI | ManBl | InfI | HWAG  | Safety | CarType   | LogisPred |
|----|--------|-------|------|-------|--------|-----------|-----------|
| 81 | 1.94   | 2.21  | 3.38 | 1.78  | -7.19  | Hatchback | Hatchback |
| 82 | -0.03  | -3.33 | 0.79 | -6.63 | 7.99   | SUV       | Hatchback |

A video player window in the bottom right corner shows a woman speaking.

Now, let us look at the result. Now we want to classify whether the test point is hatchback or SUV by setting a threshold value. So, in this case I am going to set a threshold value of 0.5.

So, now, I am going to say that from the data crashTest\_1\_test create a column call logispred and if the value that we have calculated for that point which is logispred, if that is less than or = 0.5, then assigned hatchback under this column and again from the same data if the logispred is greater than 0.5 assigned SUV under this column.

If you do that and if you run the commands this is how it creates a column. So, if you can see the last column which is the 7th column contains the predicted values and under each of these I have whether it is hatchback or SUV. Now, the reason to do this is to check how accurately our classifier is able to classify an unseen data.

(Refer Slide Time: 17:34)

The screenshot shows a RStudio session with the following content:

```
Data Science for Engineers
Confusion matrix

confusionMatrix(table(crashTest_1_TEST[,7],
crashTest_1_TEST[,6]),positive = 'Hatchback')

Confusion Matrix and Statistics
Reference
Prediction Hatchback SUV
Hatchback 10 1
SUV 0 9

Accuracy : 0.95
95% CI : (0.7513, 0.9987)
No Information Rate : 0.5
P-Value [Acc > NIR] : 2.003e-05

Kappa : 0.9
McNemar's Test P-Value : 1
```

Below the code and output, the status bar shows "Logistic Regression in R" and the number "30".

So, now let us look at the confusion matrix. So, the function is confusion matrix with a capital M, now again to use this function you should have already loaded the library caret.

Now, my input to this function is a table. So, I have the table command, now inside the table command I am giving my predicted values which is in the column 7 from the data crashTest\_1\_test, and I am giving the actual values which are the actual labels. There is also another parameter called positive. Now by default if you do not give any class as a positive class the command chooses the first class that it encounters as the positive class. So, if you do not want that you can always go back and change it under the parameter positive.

Now, I have the confusion matrix below. So, if you look at it, I have the reference labels here and I have the predicted labels here. So, this says that predicted as hatchback truly hatchback there are 10 cases, it has identified all the 10 hatchbacks correctly. But, predicted as hatchback, but truly SUV is one and predicted as SUV and truly SUV are 9. So, out of the 10 SUV cases it has identified 9 correctly and there has been 1 mis-classification.

Now, if you look at the accuracy value it is 0.95. If you can recall from Professor Raghu's performance measure lecture accuracy is nothing but the sum of the true positive and true negative divided by the total number of observations which is 20 in this case.

(Refer Slide Time: 19:21)

```
confusionMatrix(table(crashTest_1_TEST[,7],
crashTest_1_TEST[,6]),positive = 'Hatchback')
```

|                               |
|-------------------------------|
| Sensitivity : 1.0000          |
| Specificity : 0.9000          |
| Pos Pred Value : 0.9091       |
| Neg Pred Value : 1.0000       |
| Prevalence : 0.5000           |
| Detection Rate : 0.5000       |
| Detection Prevalence : 0.5500 |
| Balanced Accuracy : 0.9500    |
| 'Positive' Class : Hatchback  |

So, I again have the command on the top. I have the sensitivity value which is equal to one. The positive labels here are hatchback and all of them have been identified correctly, whereas if you can see there has been one misclassification and hence this specificity drops to 0.9. There is also something called balanced accuracy which = 0.95. Now, balanced accuracy is the average of sensitivity and specificity.

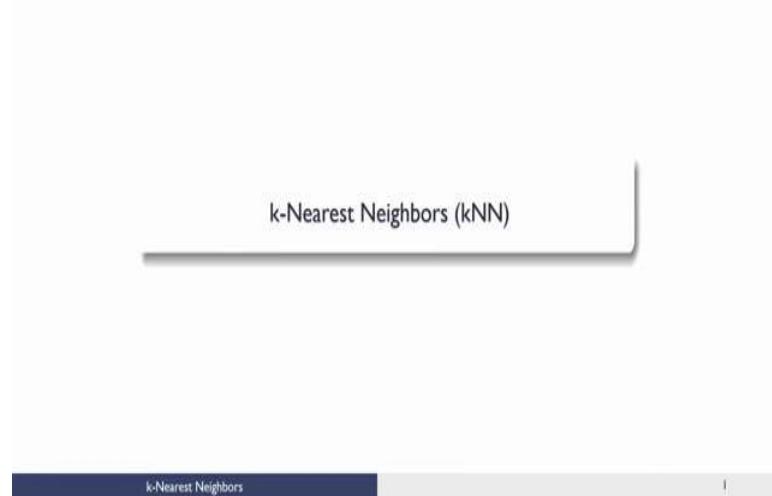
All the other performance measures have been explained by Professor Raghu in his lecture of performance measure and you can always go back and refer to it to know more about the other performance measures. In this lecture we looked at the case study of automotive crash testing we also saw how to read the data, we saw how to understand it, we used `glm` function to model logistic regression and we looked at using confusion matrix to interpret the results.

Thank you.

**Data science for Engineers**  
**Prof. Ragunathan Rengasamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 46**  
**K - Nearest Neighbors (kNN)**

(Refer Slide Time: 00:12)



In this lecture, we will look at a very very simple yet powerful classification algorithm called the k nearest neighbors. So, let me introduce a k nearest neighbor classification algorithm.

(Refer Slide Time: 00:30)

- k Nearest Neighbors(kNN) is a non-parametric method used for classification
- It is a lazy learning algorithm where all computation is deferred until classification
- It is also an instance based learning algorithm where the function is approximated locally



It is, what is called a nonparametric algorithm for classification. So, let me explain what this non parametric means. Remember when we looked at logistics regression for example, we said, let us say there is data like this and we are going to use the trained data, to develop a hyperplane of this form  $\beta_0 + \beta_{11} X_1 + \beta_{12} X_2$  in the 2 d case. And any time a new data point comes, what we do is, we use the parameters that have been estimated from the train data to make predictions about test data.

So, remember we had this  $e$  power this term divided by  $1 + e$  power this term right here. So, this function is actually a function of  $\beta_0$ ,  $\beta_{11}$  and  $\beta_{12}$ . So, these are all parameters that have already been derived out of this data. And any time a new test data comes in, it is sent through this function and then you make a prediction. So, this is a parametric method, because parameters have been derived from the data. k nearest neighbor is a different idea, where I do not get parameters like this out of the data. I am going to use the data itself to make classifications. So, that is an interesting different idea that one uses in k nearest neighbors.

I just want to make sure that we get the terminology right. We will later see that the k nearest neighbor, there is one parameter that we use for classifying, which is the number of neighbors that I am going to look at. So, I do not want you to wonder, since we are using anyway a parameter in this k nearest neighbor why am I calling it nonparametric. So, the distinction here is subtle, but I want you to remember this. The number of neighbors that we use in the k nearest neighbor algorithm that you will see later, is actually a tuning parameter for the algorithm, that is not a parameter that I have derived from the data.

Whereas, in logistics regression, these are parameters I can derive only from the data. I cannot say what these values will be a priori.

Whereas, I could say, I will use a k nearest neighbor with two neighbors three neighbors and so on, so that is a tuning parameter. So, I want you to remember the distinction between a tuning parameter and parameters that are derived from the data and the fact that k nearest neighbor is a nonparametric method, speaks to the fact that we are not deriving any parameters from the data itself. However, we are free to use tuning parameters for k nearest neighbors, so that is an important thing to remember. It is also called a lazy learning algorithm, where all the computation is deferred until classification.

So, what we mean by this is the following. If I give trained data for example, for logistics regression. I have to do this work to get these parameters, before I can do any classification for a test data point . So, without these parameters I can never classify test data points. However, in k nearest neighbor it just give me data and a test data point I will classify.

So, we will see how that is done, but no work needs to be done before I am able to classify a test data point. So, that is an other important difference between k nearest neighbor and logistic regression for example. It is also called as an instant based learning where the function is approximated locally. So, we will come back to this notion of local as I describe this algorithm.

(Refer Slide Time: 04:37)

Data science for Engineers

## Why kNN and when does one use it?

- Why kNN ?
  - Simplest of all classification algorithms and easy to implement
  - There is no explicit training phase and the algorithm does not perform any generalization of the training data
- When does one use this algorithm?
  - When there are nonlinear decision boundaries between classes
  - When the amount of data is large



k-Nearest Neighbors

Now we may ask, when do we use this. As I started this lecture I mentioned it simplest of classification algorithms, very easy to implement and you will see when I explain the algorithm how simple it is. There is no explicit training phase and so on and there is no generalization for the training data and all that. It is just that I give the data and then I just wait till they give me a new data point, to say what class it should belong to. Of course, based on the algorithm itself I can

also predict for the train data points itself what class they should belong to and then maybe compare it with the label that the data point has and so on.

Nonetheless I am not going to explicitly get some parameters out. And when does one use this algorithm, this is a simple algorithm when there are complicated non-linear decision boundaries, this algorithm actually works surprisingly well, and when you have large amount of data and the train phase can be bogged down by large number of data in terms of an optimization algorithm and so on then you can use this. However, a caveat is you will see as we describe this algorithm when you have more and more data, the classification of nearest neighbor itself, will become complicated.

So, there are ways to address this, but when we say, when the amount of data is large, all that we are saying is since there is no explicit training phase, there is no optimization with a large number of data points, to be able to identify parameters that are useless at later in classification. So, in other words, in other algorithms you will do all the effort a priori and once you have the parameters then classification becomes, on the test data point becomes, easier. However, since kNN is a lazy algorithm all the, all the calculations are deferred till you had actually have to do something, at that point there might be lot more classic, lot more calculations if the data is large.

(Refer Slide Time: 06:54)

Data science for Engineers

## k Nearest Neighbors

- Input features
  - Input features can be both quantitative and qualitative
- Outputs
  - Outputs are categorical values, which typically are the classes of the data
- kNN explains a categorical value using the majority votes of nearest neighbors



k-Nearest Neighbors

So, the input features for k nearest neighbors could be both quantitative and qualitative, and output are typically categorical values which are what type of class does this data belong to. Now it is not necessary that we use k nearest neighbor only for classification though

that is where it's used the most. You could also use it with very simple extensions or simple definitions for function approximation problems also, and you will see as I describe this algorithm how it could be adapted for function approximation problems quite easily.

Nonetheless as far as this lecture is concerned, we are going to say the outputs or categorical values, which basically says different classes and what class does this data point belong to. In one word if you were to explain k nearest neighbor algorithm, you would simply say k nearest neighbor explains the categorical value, using the majority votes of nearest neighbors.

So, what basically we are saying is, if there is a particular data point and I want to find out which class this data point belongs to, all I need to do is look at all the neighboring data points and then find which class they belong to and then take a majority vote and that is what is the class that is assigned to this data point. So, it's something like if you want to know a person, you know his neighbors, something like that is what use using k nearest neighbors.

(Refer Slide Time: 08:36)

Data science for Engineers

## Assumptions

- Being nonparametric, the algorithm does not make any assumptions about the underlying data distribution
- Select the parameter  $k$  based on the data
- Requires a distance metric to define proximity between any two data points
  - Example: Euclidean distance, Mahalanobis distance or Hamming distance

0:00 / 0:00   k-Nearest Neighbors

Now remember at the beginning of this portion of data science algorithms I talked about the assumptions that are made by different algorithms. Here for example, because this is a nonparametric algorithm, we really do not make any assumptions about the underlying data distribution. We are just going to look at the nearest neighbors and then come up with an answer. So, we are not going to assume probability distribution or any other linear separability assumptions and so on.

As I mentioned before, this  $k$ , the number of neighbors we are going to look at, is a tuning parameter and this is something that you select. So, you use a tuning parameter, run your algorithm and you get good results, then keep that parameter if not you kind of play around with it and then find the best  $k$  for your data. The key thing is that because we keep talking about neighbors, and from a data science viewpoint whenever we talk about neighbors, we have to talk about a distance between a data point and its neighbor.

We really need a distance metric for this algorithm to work and this distance metric would basically say what is the proximity between any two data points. The distance metric could be Euclidean distance, Mahalanabis distance, Hamming distance and so on. So, there are several distance metrics that you could use to basically use  $k$  nearest neighbor.

(Refer Slide Time: 10:14)

**Data science for Engineers**

## Algorithm

- The kNN classification is performed using the following four steps
  - Compute the distance metric between the test data point and all the labeled data points
  - Order the labeled data points in the increasing order of this distance metric
  - Select the top  $k$  labeled data points and look at the class labels
  - Find the class label that the majority of these  $k$  labeled data points have and assign it to the test data point

**k-Nearest Neighbors**

So, in terms of the algorithm itself it is performed using the following four steps. Nothing is done till the algorithm gets a data point to be classified. Once you get a data point to be classified, let us say I have  $N$  data points in my database and each has a class label. So for example,  $X_1$  belongs to class 1,  $X_2$  belongs to class 1,  $X_3$  belongs to class 2 and so on,  $X_n$  belongs to let us say class 1. So, this is you know a binary situation, binary classification problem. This need not be a binary classification problem; for example,  $X_2$  could belong to class 2 and so on.

So, there might be many classes. So, multi class problems are also very very easy to solve using kNN algorithm. So, let us anyway stick to binary problem. Then what you are going to do is, let us say I have a new test point which I call it  $X_{\text{new}}$  and I want to find out how I classify this. So, the very first step which is what we talk about here, is we find a

distance between this new test point and each of the labelled data points in the data set. So for example, there could be a distance  $d_1$  between  $X_v$  and  $X_1$ ,  $d_2$  between  $X_v$  and  $X_2$ ,  $d_3$  and so on and  $d_n$ . So, once you calculate this distance, then what you do is you have  $n$  distances and this is the reason why we said you need a distance metric in last slide for a kNN to work.

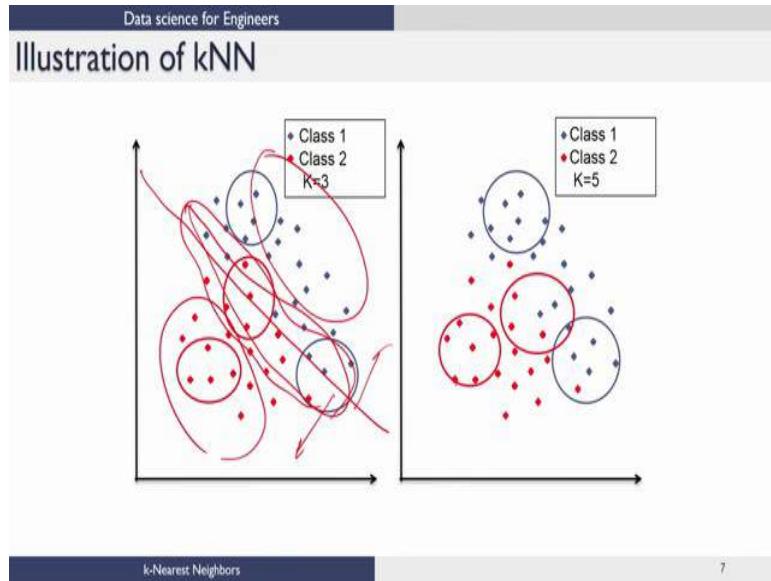
Once we have this distance then what we do, is basically we look at all of these distances and then say, I order the distances from the smallest to the largest. So, let us say if  $d_n$  is the smallest distance, so  $d_n$  maybe  $d_5$ ,  $d_3$ ,  $d_{10}$ , whatever it is, so I order them. This is the smallest to the largest. You can also think of this as closest to the farthest, because the distances are all from  $X_{new}$ .

So, the distance is zero then the point is  $X_v$  itself. So, any small distance is the closest to  $X_{new}$  and as you go down it is further and further. Now the next step is very simple. If let us say you are looking at  $k$  nearest neighbors with  $k = 3$ , then what you are going to do, is you are going to find the first three distances in this and this distance is from  $X_n$ , this distance is from  $X_5$  and this distance is from  $X_3$ .

So, once we order this according to distance and go from the smallest to largest, once we sort it in this fashion, then we also know what the corresponding data points are. So, this belongs, this is the data point  $X_n$ , so this is the distance between  $X_n$  and  $X_{new}$ , distance between  $X_5$  and this. So, now, I have these three data points that I picked out from the data set. Now if I want to classify this all that I do is the following, I find out what class these data points belong to. So, if all of them belong to class 1, then I say  $X_{new}$  is class 1. If all of them belong to class 2, I say  $X_{new}$  is class 2. If two of them belong to class 1 and the third one belongs to class 2, I do a majority vote and still say its class 1. If two of them belong to class 2 and one belongs to class 1 I say its class 2 that is it, that is all the algorithm is.

So, it says to find the class label that the majority of this  $k$  label data points have and I assign it to the test data point, very simple. Now I also said this algorithm with minor modifications can be used for function approximation. So, for example, if you so choose to, you could take this and then let us say if you want to predict what an output will be for a new point, you could find the output value for these three points and take an average. For example, very trivial, and then say that is the output corresponding to this, so that becomes of adaptation of this for function approximation problems and so on. Nonetheless for classification this is the basic idea. Now if you said  $k = 5$  then what you do is, you go down to 5 numbers and then do the majority vote, so that is all we do. So, let us look at this very simple idea here.

(Refer Slide Time: 15:05)



Let us say this is actually the training data itself and then I want to look at  $k = 3$  and then see what labels will be for the training data itself. The blue are actually labeled, so this is supervised. So, the blue are all belonging to class 1 and the red is all belonging to class 2, and then let us say for example, I want to figure out this point here blue point. Though I know the label is blue, what class would  $k$  nearest neighbor algorithm say this point belongs to. Say if I want to take  $k = 3$ , then basically I have to find three nearest points which are these three, so this is what is represented.

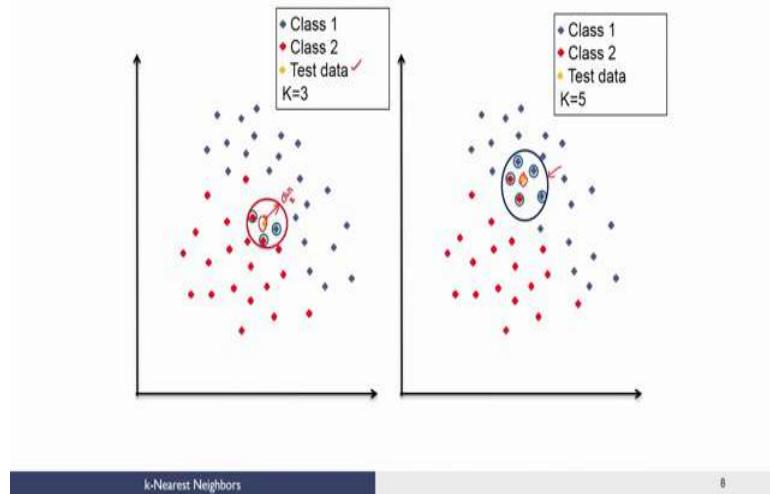
And since the majority is blue, this will be blue. So, basically if you think about it, this point would be classified correctly and so on. Now even in the training set for example, if you take this red point, I know the label is red; however, if I were to run  $k$  nearest neighbor with three data points, when you find the three closest point, they all belong to blue. So, this would be misclassified as blue even in the training data set. So, you will notice one general principle is, there is a possibility of data points getting misclassified, only in kind of this region where there is a mix of both of these data points.

However, as you go further away, the chance of miss classification keeps coming down. So, again in some sense you see a parallel, saying if I were to draw a line here and then say this is all red class, this is all blue class. Then points around this line is where the problem is, as they go further away the problems are less. Nonetheless notice how we have never defined a boundary line or a curve here at all, the data points themselves tell you how this boundary is drawn. So, in that sense, its, while it is simple it is also in something sophisticated, because we never have to guess a boundary, the data points themselves define a boundary in some sense. So, I can actually effectively use this

algorithm for complicated non-linear boundaries, which you would have to guess a priori if we were using a parametric approach, so that is a key idea here. a similar illustration for  $K = 5$ .

Now if I want to let us say a check this data point from the training set itself, then I look at its five neighbors, closest 1 2 3 4 5, all of them are red. So, this is classified as red and so on. So, this is the basic idea.

(Refer Slide Time: 18:06)



Now, you do not have to do anything till you get a data point. So, you could verify how well the algorithm will do on the training set itself. However, if you give me a new test data here, so which is what you shown by this data point. Then if you want to do a classification there is no label for this. Remember the other red and blue data points already have a label from prior knowledge, this does not have a label.

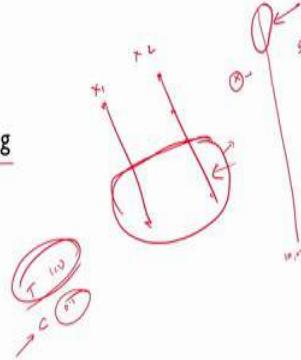
So, I want to find out a label for it. So, if I were to use  $k = 3$ , then for this data point I will find the three closest neighbors, they happen to be these three data points. Then I will notice that two out of these are red, so this point will get a label + 2. If on the other hand the test data point is here and you were using  $K = 5$  then, you look at the 5 closest neighbor to this point and then you see that two of them are class 2 and three are class 1, so majority voting, this will be put into class 1. So, you will get a label of class 1 for this data point. So, this is the basic idea of  $k$  nearest neighbor, so very very simple.

(Refer Slide Time: 19:17)

Data science for Engineers

## Things to consider

- Following are some things one should consider before applying kNN algorithm
  - Parameter selection
  - Presence of noise
  - Feature selection and scaling
  - Curse of dimensionality



k-Nearest Neighbors

However, these are some of these the things to consider before applying kNN. So, one has to choose the number of neighbors that one is going to use, the value of  $k$ , whether its 3 5 7 9 whatever that is, and the results can quite significantly depend on the parameter that you choose. Particularly when you have noise in the data and that has to be taken into account. The other thing to keep in mind when using kNN is that, when you do a distance between two data points  $X_1$  and  $X_2$  let us say, and let us say there are  $n$  components in this, the distance metric will take all of these components into picture.

So, since we are comparing distance, then that basically means every attribute for this data point and this data point we are comparing distances. The problem with this is that, if for example, there are a whole lot of attributes which actually are not at all important from a classification viewpoint, then what happens is, though they are not important from a classification viewpoint, they contribute in the distance measure. So, there is, there is a possibility of these features actually kind of spoiling the results in  $k$  nearest neighbor.

So, it is important to pick features which are which are of relevance in some sense, so the distance metric actually uses only features which will give it the discriminating capacity. So, that is one thing to keep in mind. The other the problem is, these are these are handle able, these are rather easily handled, but these are things to keep in mind when you look at these kinds of algorithms and also particularly this being the first course on data science I am assuming for most of you, these are kinds of things that you might not have thought about before.

So, it's worthwhile to kind of think about this, do some mental experiments to see why these kinds of things might be important and so on. Now the other aspect is scaling. So, for example, if there are two attributes, let us say in data temperature and concentration, and temperatures are in values of 100, concentrations are in values of 0.1 0.2 and so on. When you take a distance measure and these numbers will dominate over this.

So, it is always a good idea to scale your data in some format before doing this distance. Otherwise while this might be an important variable from a classification viewpoint, it will never show up, because these numbers are bigger and they will simply dominate the small number. So, feature selection and scaling are things to keep in mind. And the last thing is curse of dimensionality. So, I told you that while this is a very nice algorithm to apply, because there is not much computation that is done at the beginning itself. However, if you notice, if I get a test data point and I have to find let us say the 5 closest neighbor, there is no way in which I can do this, it looks like, unless I calculate all the distances.

So, that can become a serious problem, if the number of data points in my database is very large. Let us say I have 10000 data points, and let us assume that I have an algorithm k nearest neighbor algorithm with  $K = 5$ . So, really what I am looking for, is finding 5 closest data points from this data base to this data point. However, it looks like I have to calculate all the 10,000 distances and then sort them and then pick the top 5. So, in other words to get this top 5 have to do so much work. So, there must be smarter ways of doing it, but nonetheless one has to remember the number of data points and number of features, one has to think how to apply this algorithm carefully.

(Refer Slide Time: 24:00)

Data science for Engineers

## Parameter selection

- The best choice of  $k$  depends on the data
- Larger values of  $k$  reduce the effect of noise on classification but makes the decision boundaries between classes less distinct
- Smaller values of  $k$  tend to be affected by the noise with clear separation between classes



So, the best choice of  $k$  depends on the data and one general rule of thumb is, if you use large values for  $k$ , then clearly you can see you are taking lot more neighbors, so you are getting lot more information. So, the effect of noise on classification can become less. However, if you take large number of neighbors, then your decision boundaries are likely to become less crisp and more diffuse.

So, because if let us say there are two classes like this, then for this data point if you take a large number of neighbors, then you might pick many neighbors from the other class also, so that can make the boundaries less crisp and more diffuse. On the fifth slide, flipside, if you use smaller values of  $k$  then your algorithm is likely to be affected by noise and outliers, however, your decision boundaries as a rule of thumb are likely to become crisper. So, this is, these are some things to keep in mind.

(Refer Slide Time: 25:08)

Data science for Engineers

## Feature selection and scaling

- It is important to remove irrelevant features
- When the number of features is too large, and suspected to be highly redundant, feature extraction is required
- If the features are carefully chosen then it is expected that the classification will be better

And as I mentioned before it is important to remove irrelevant features and scaling is also an important idea. So, if you choose your features carefully, then you would get better classification with kNN. So, with this we come to an end of this lecture on  $k$  nearest neighbors and following this lecture there will be a case study, which will use  $k$  nearest neighbor that will be taught by one of the teaching assistants. And after that I will teach a lecture on  $k$  means clustering. Thank you and I look forward to seeing you again in a future lecture.

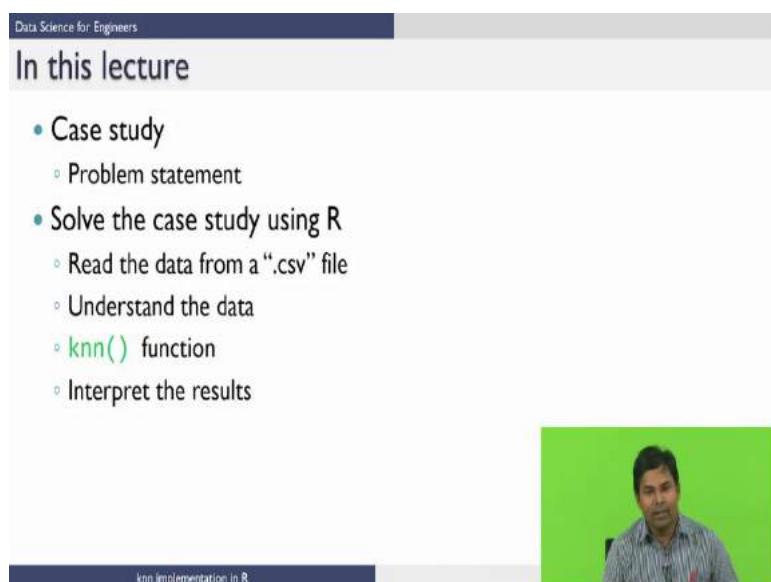
Thanks.

**Data science for Engineers**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 47**  
**K- nearest neighbours implementation in R**

Hello all, welcome to this lecture on K-nearest neighbours implementation in R.

(Refer Slide Time: 00:23)



The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data Science for Engineers'. Below the header is a light grey section with the title 'In this lecture' in bold black font. Underneath the title is a bulleted list of topics:

- Case study
  - Problem statement
- Solve the case study using R
  - Read the data from a ".csv" file
  - Understand the data
  - `knn()` function
  - Interpret the results

At the bottom of the slide, there is a dark blue footer bar with the text 'knn implementation in R'.

In the top right corner of the slide area, there is a small video frame showing a man with dark hair and a striped shirt, likely the lecturer, sitting in front of a green screen.

In this lecture, what we are going to do is to introduce you to a case study which we use as a means to explain how to implement this knn algorithm in R. We will start with the problem statement of the case study and we will show how to solve this case study using R.

In the process we will show how to read the data from dot csv le, how to understand the data that is being loaded into the workspace of R and how to implement this K-nearest neighbours algorithm in R using this knn function. And we will also talk about how to interpret the results that this knn algorithm gives to us.

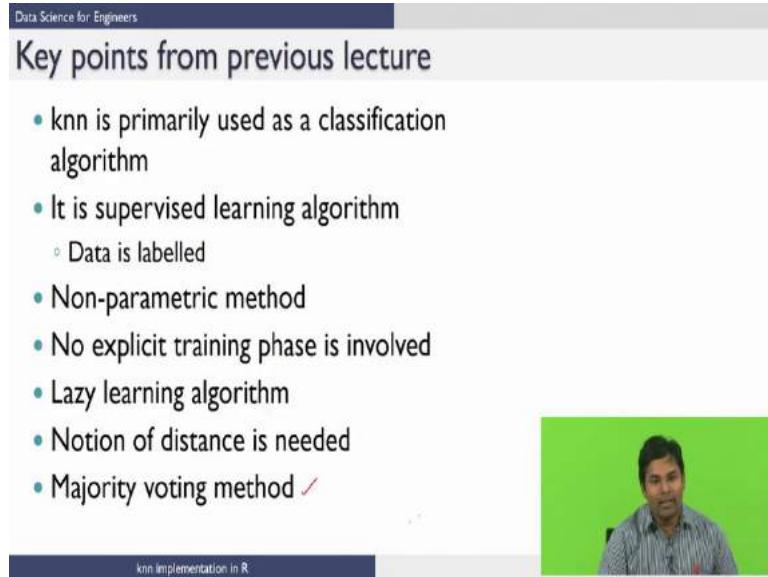
(Refer Slide Time: 01:12)

Data Science for Engineers

## Key points from previous lecture

- knn is primarily used as a classification algorithm
- It is supervised learning algorithm
  - Data is labelled
- Non-parametric method
- No explicit training phase is involved
- Lazy learning algorithm
- Notion of distance is needed
- Majority voting method ✓

knn implementation in R



Before we jump into the case study, let us review some key points from the previous lecture of Prof. Raghu. If you remember knn is primarily used as a classification algorithm. It is a supervised learning algorithm. When I say supervised learning algorithm that means the data that is provided to you has to be labelled data and knn is a non-parametric method. So, what do you mean by this non-parametric method is that there is no extraction of the parameters of the classifiers from the data itself. And there is no explicit training phase involved in this knn algorithm.

And the knn algorithm is a lazy learning algorithm because it would not do any computations till you ask you to do classification. Because we are dealing with the K-nearest neighbours we would have seen this notion of distance is important when we are dealing with this knn algorithm. And the way the knn algorithm works is by the majority voting method. That means if you give a test point, we calculate the distance of the test point from all the data points in the given data and arrange them in the ascending order and we choose the k first nearest neighbours. And based on the voting that each of them will give for this test data, we will assign the class to the test data point. That is how essentially the knn works.

Now, let us define the case study problem statement. We have named this case study as automotive service company case study.

(Refer Slide Time: 03:07)

Data Science for Engineers

### Automotive Service Study: Problem statement

An automotive service chain is launching its new grand service station this weekend. They offer to service a wide variety of cars. The current capacity of the station is to check 315 cars thoroughly per day.

As an inaugural offer, they claim to freely check all cars that arrive on their launch day, and report whether they need servicing or not!

Unexpectedly, they get 450 cars. The service men won't work longer than the working hours but the data analysts have to!

Can you save the day for the new service station?



Let us look at the problem statement. An automotive service chain is launching its grand new service station this weekend. They offer service to wide variety of cars. The current capacity of the station is to check the 315 cars thoroughly per day. As an inaugural offer, what they have done is, they claim to freely check all the cars that arrive on their launch day and they said they will report whether they need servicing or not.

What happened is unexpectedly, they got 450 cars. Now, since they have the testing facility for testing only 315 cars, they will not be able to check all the 450 cars very thoroughly and the servicemen will not work longer than the normal working hours.

So, what they have done is they have hired a data analyst to help them out from the situation. If you are the data analyst which is hired by this automotive service station person, how can you save the day for this new service station is the problem statement.

(Refer Slide Time: 04:34)

Data Science for Engineers

## How can a data scientist save a day for them?

- He has been a data set which contains some attributes of car that can be easily measured and wont require much time and a conclusion that if service is needed for that or not. - "serviceTrainData.csv" ✓
- Now for the cars they cannot check in detail, they measure those attributes- "serviceTestData.csv" ✓
- Use knn classification technique to classify the cars they cannot test manually and say whether service is needed or not .

knn implementation In R



Now, let us see how a data scientist can save a day for this service station people. Since, service station has capacity to thoroughly check 315 cars, they have thoroughly checked all the 315 cars and given the data in this service traindata.csv. Now, for the rest of the cars among the 450, they cannot thoroughly check all the data and they have checked only those attributes which are easily measurable and they have given them in this service test data dot csv. So, essentially the data scientist has data which is like a training data for him which contains few attributes and with a label whether a service is needed or not.

And he also has a data for which now all the other attributes are present, he do not have this column where whether the service is needed or not. The idea here is how do one use this data, service train data, to comment upon for the readings which present which are present in the service test data to tell whether service is needed or not in this case. So, the idea is to use this knn classification technique to classify the cars in the service test data le which cannot be tested manually and say whether service is needed or not. Now, let us see how do you solve this case study in R.

(Refer Slide Time: 06:26)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data Science for Engineers'. Below the header, the main title 'Getting things ready' is displayed in a large, bold, black font. To the right of the title, there is a small video frame showing a man with dark hair and a beard, wearing a striped shirt, speaking. The video frame has a green background. On the left side of the slide, there is a code block in R syntax. The code is as follows:

```
knn Implementation in R
Set the working directory as the directory which contains the
data files
setwd("Path of the directory with data files")
rm(list=ls()) # to clear the environment
install.packages("caret",dependencies = TRUE)
install.packages("class",dependencies = TRUE)
library(caret) # for confusionMatrix
library(class) # for knn
```

Below the code block, there is a dark blue footer bar with the text 'knn implementation in R'.

First you have to get things ready. When I say get things ready I mean you have to set the working directory as the directory in which the given data files are available. That you can do using set working directory command and the corresponding path you can give here. Otherwise you can use GUI option also to set the working directory. And this command here is used to clear all the variables in the environment of R. You can very well use the brush button in the environmental history pan to clear the variables in the workspace.

And another important thing one has to do is, for this knn implementation, we need two external packages which are caret and class, one has to install this caret and class packages if they have not installed it already. So, the way to install this packages we have explained in our R modules, you can install the packages through the command window using this command install dot pack-ages and the package name and say dependencies = true or you can use the GUI to install the packages. So, please install this packages caret and class. And once you install, you can load those packages using the library command as we have explained already. We will see why is this packages important as we go along this lecture.

And library caret is for generating the confusion matrix which Prof. Raghu would have talked about when he is talking about this performance matrix of a classifier. And this library class is a library which contains different classification algorithms. And here we are going to use it for implementing this knn.  
Now, let us see how to read the data.

(Refer Slide Time: 08:32)

Data Science for Engineers

## Reading the data

- Data for this case study is provided to you in files with names “serviceTrainData.csv”, “serviceTestData.csv”
- To read the data from a “.csv” file we use `read.csv()` function



knn implementation in R

From the given les and for this case, a data is being provided in two les as we have already seen servicetraindata.csv and service test data dot csv. So, in order to read this data from the csv files, function we use is read.csv function. Let us look what this read dot csv function takes and what it returns.

(Refer Slide Time: 08:59)

Reads a file in table format and creates a data frame from it

SYNTAX

`read.csv(file, row.names)`

|                        |                                                                                                                                                                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>file</code>      | the name of the file which the data are to be read from. Each row of the table appears as one line of the file.                                                                                                                            |
| <code>row.names</code> | a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names. |



This read dot csv file reads a file in a table format and creates a data frame from it. The syntax for this read dot csv function is as follows, read dot csv the filename, and the row names. Let us look at what this input arguments le and row dot names means. File is essentially the name of the file from which you have to read the data. And row dot names is a vector of row names, it can be either a vector giving the actual row names or a single value which specifies what column of the

data set is having the row names. Let us see how to read the data in this particular case.

(Refer Slide Time: 09:48)

The screenshot shows the RStudio interface. At the top, there's a header bar with 'Data Science for Engineers'. Below it, a title 'Reading the data' is displayed. A bulleted list provides instructions: 'Data for this case study is provided to you in files with names "serviceTrainData.csv", "serviceTestData.csv"'. A code block shows the R commands: '#Reading the data', 'ServiceTrain <- read.csv("serviceTrainData.csv")', and 'ServiceTest <- read.csv("serviceTestData.csv")'. To the right of the code block, there's a video player window showing a person speaking. The RStudio environment pane shows 'ServiceTest 135 obs. of 6 variables' and 'ServiceTrain 315 obs. of 6 variables'. The bottom of the screen has a dark bar with the text 'knn implementation in R'.

As we have seen the data has been given in this two dot csv files, we can use read dot csv function to read the data. As we have seen in the syntax of read.csv we have to give the filename that is the filename service train dot data from which I want to load the data. I will give this file name. And I am assigning this two a variable called service train when you execute this command what happens is, it reads a data from the service train data file and assign it to this variable which is of the form data frame.

Similarly, you will read the data from service test data and assign it to variable service test which is again a data frame. In the R environment, once you execute these commands you will see two data frames which are service test and service train which are having this 315 observations of 6 variables and 135 observations of 6 variables.

Remember why this 315? 315 is the number of cars that they can thoroughly check, but they have given in this 315 the 6 variables are the attributes which are easily measurable and one column which says whether service is needed or not. And this 135 cars they have 6 variables, they have measured all the 5 attributes which are important and the 6 attribute is also given here we will see why the 6 attribute is given and so on as we go on in this lecture.

Now, let us see what is there in this service train and service test data. One way to see what is there in this service test and service train is to use the view command.

(Refer Slide Time: 11:42)

|    | Cylinders  | EnginePerf | NormMileage | TyreWear   | HVACwear     | Service |
|----|------------|------------|-------------|------------|--------------|---------|
| 1  | 41.773338  | 49.936015  | 49.775881   | 48.263851  | 50.95207173  | No      |
| 2  | 4.987115   | 7.891033   | 6.518986    | 9.493161   | 3.24026218   | No      |
| 3  | 4.987115   | 4.891033   | 7.318986    | 8.373161   | 2.78026218   | No      |
| 4  | 106.388821 | 104.454032 | 103.051485  | 106.282658 | 105.53564290 | No      |
| 5  | 104.388821 | 101.744032 | 103.051485  | 106.132658 | 105.77064290 | No      |
| 6  | 4.987115   | 8.891033   | 5.018986    | 8.373161   | 1.78026218   | No      |
| 7  | 45.535338  | 58.666615  | 48.147581   | 50.033851  | 47.35207173  | No      |
| 8  | 27.705516  | 28.193859  | 31.259358   | 31.225162  | 31.31127500  | Yes     |
| 9  | 28.705516  | 28.418205  | 30.809536   | 29.200162  | 31.31127500  | Yes     |
| 10 | 104.388821 | 101.744032 | 105.051485  | 106.212058 | 104.24684290 | No      |
| 11 | 4.987115   | 5.891033   | 7.228986    | 8.373161   | 1.08026218   | No      |
| 12 | 104.388821 | 101.434032 | 104.051485  | 106.002658 | 105.53564290 | No      |

This view command helps you to see the data frames. For example, if you want to see what is there in the service train data frame, what you have to do is this view service train will show a table like this in your editor environment. Now, you can see that there are how many attributes 1, 2, 3, 4, 5, 6 attributes. And if you see these are the five attributes which are measured for testing whether the service is needed or not, and this attribute is basically saying if service is needed or not.

Similarly, you can see for the service test data set which is shown here. For now, what we assume is will act such a way that we do not know this column and we will come back to this. Now, if you observe here, there are 135 entries for which they have not thoroughly checked they just measured this 6 quantities and they want to figure out whether service is needed or not using the knn algorithm that is the whole idea. Since, you have viewed what is there in this service test and service train data sets, now is there any way to know what are the data types of the these attributes that are there in this service train and service test is the next question that comes to mind. Now, let us understand the data and little more detail

(Refer Slide Time: 13:12)

Data Science for Engineers

## Understanding the data

- ServiceTrain contains 315 observations of 6 variables
- ServiceTest contains 135 observations of 6 variables
- The variables are: OilQual, Engineperf, NormMileage, TyreWear, HVACwear and Service
  - First five columns are the details about the car and last column is the label which says whether a service is needed or not

knn implementation in R



What we have seen till now is the service train contains 315 observations of six variables, service test contains 135 observations in 6 variables. And variables that are present in the data sets are oil quality, engine performance, normal mileage, tyre wear, HVAC wear and service. And I as I mentioned earlier this 5 are the attributes that tells about the condition of the car. And this attribute simply says whether service is needed or not that is what here.

First five columns are the details about the car and the last column is the label which says whether a service is needed or not. Now, let us ask this question what are the data types of each of these attributes, how one get the data types of the attributes that are there in the data.

So, since we have understood the data now. Let us look at what is the structure of the data.

(Refer Slide Time: 14:11)

Data Science for Engineers

## Structure of the data

- Structure of data
  - Variables and their data types
- **str()**  
Compactly display the internal structure of an R object

SYNTAX

**str(object)**

object      any R object about which you want to have some information.

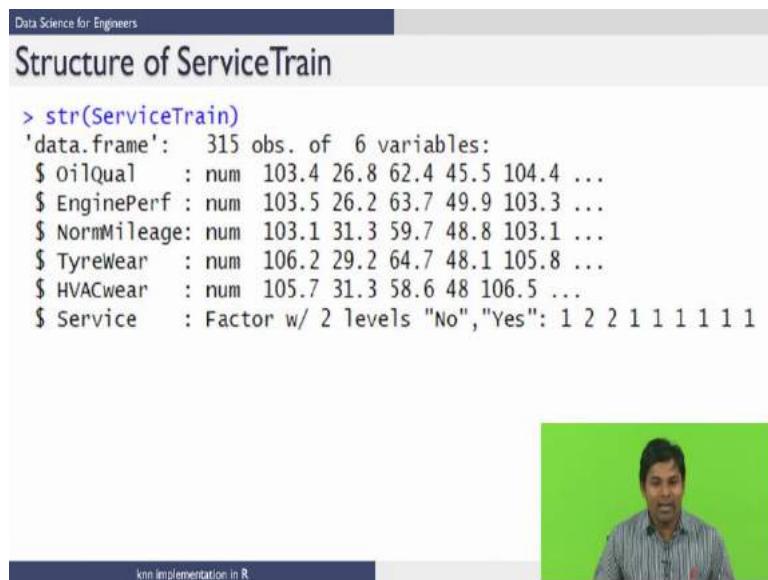
knn implementation in R



When you say structure of data what do we mean by that is in the data set you have what are the variables that are there, and what are their data types. So, the way you get the structure of data in R is using this structure function. What does this structure function do, structure function compactly display the internal structure of an R object. The syntax for the structure function is as follows. Structure function takes one input argument which is an object. What is this object, this object is essentially any R object about which you want to have some information.

Now, let us see the structure of two data frames what we have read from the two dots csv files.

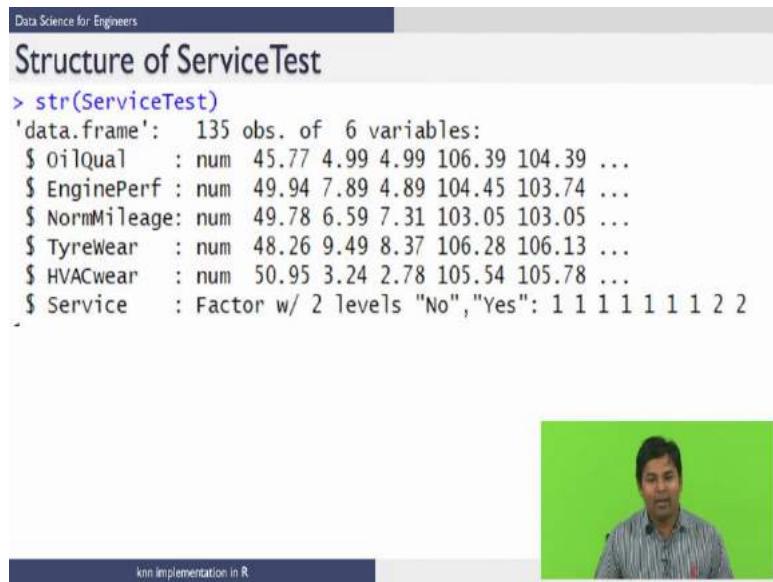
(Refer Slide Time: 14:58)



```
> str(ServiceTrain)
'data.frame': 315 obs. of 6 variables:
 $ OilQual : num 103.4 26.8 62.4 45.5 104.4 ...
 $ EnginePerf : num 103.5 26.2 63.7 49.9 103.3 ...
 $ NormMileage: num 103.1 31.3 59.7 48.8 103.1 ...
 $ TyreWear : num 106.2 29.2 64.7 48.1 105.8 ...
 $ HVACwear : num 105.7 31.3 58.6 48 106.5 ...
 $ Service : Factor w/ 2 levels "No","Yes": 1 2 2 1 1 1 1 1 1
```

You can see the structure of the service train data frame. Here, if you execute this command structure of service train, what it gives is the following information which says service train is a data frame which contains 315 observations of six variables. And the variables are oil quality, engine performance and so on. And they will say the data type of all this five attributes is numeric, and the last attribute service is a factor with two levels that means we have yes or no in this attribute. And this one two represents each entry for example one corresponds to no and two corresponds to yes and so on. Let us use the structure command on the service test data and see what it has.

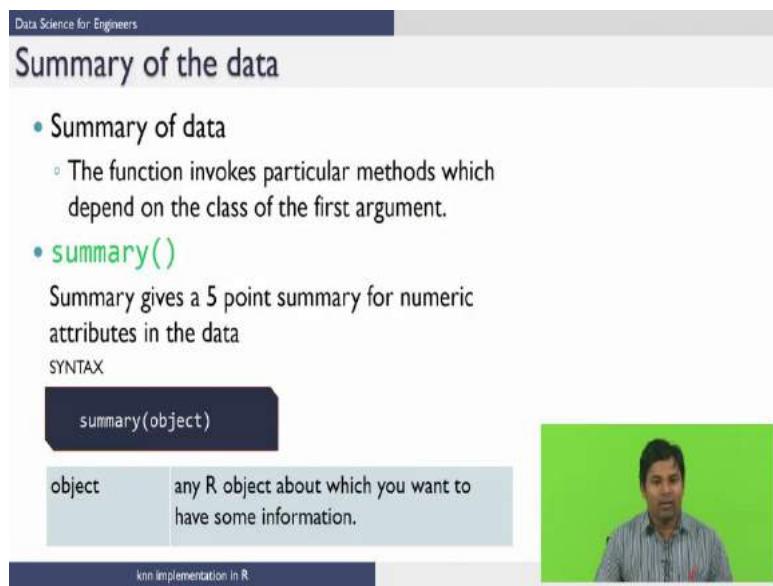
(Refer Slide Time: 15:54)



```
> str(ServiceTest)
'data.frame': 135 obs. of 6 variables:
 $ oilQual : num 45.77 4.99 4.99 106.39 104.39 ...
 $ EnginePerf : num 49.94 7.89 4.89 104.45 103.74 ...
 $ NormMileage: num 49.78 6.59 7.31 103.05 103.05 ...
 $ TyreWear : num 48.26 9.49 8.37 106.28 106.13 ...
 $ HVACwear : num 50.95 3.24 2.78 105.54 105.78 ...
 $ Service : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2
```

This is the output you see when you execute this command here. It says the service test is also data frame which contains 135 observations in 6 variables. These are the variables that are available. The first 5 variables are numeric type variables, and the service variable is a factor with two levels which contains yes or no.

Since we have seen the structure of the data, let us ask this question is there any way that I will get a summary of the data which I have read.  
(Refer Slide Time: 16:27)



- Summary of data
  - The function invokes particular methods which depend on the class of the first argument.
- **summary()**

Summary gives a 5 point summary for numeric attributes in the data

SYNTAX

```
summary(object)
```

|        |                                                             |
|--------|-------------------------------------------------------------|
| object | any R object about which you want to have some information. |
|--------|-------------------------------------------------------------|

The answer is yes, you can get. The summary of data is obtained by the summary function. Essentially what it does is it invokes particular methods depending upon the class of the argument that goes along with this summary function. For example, summary function gives a 5 point summary for numeric attributes in the data. Syntax for the summary function is as follows. The summary function takes one argument

which is an object. This object is any R object about which you want to get some information.

Let us use the summary function on our data frames which we have loaded and see what the results are.

(Refer Slide Time: 17:15)

Data Science for Engineers

## Summary of ServiceTrain

```
> summary(ServiceTrain)
 OilQual EnginePerf
Min. : 0.9872 Min. : 1.891
1st Qu.: 26.7655 1st Qu.: 27.418
Median : 59.6633 Median : 59.741
Mean : 59.6493 Mean : 60.306
3rd Qu.:104.3888 3rd Qu.:103.744
Max. :106.4288 Max. :105.744
 NormMileage TyreWear
Min. : 3.359 Min. : 6.213
1st Qu.: 31.260 1st Qu.: 29.036
Median : 57.221 Median : 60.304
Mean : 60.297 Mean : 61.759
3rd Qu.:103.051 3rd Qu.:106.173
Max. :105.051 Max. :108.173
 HVACwear Service
Min. : -1.72 No :232
1st Qu.: 31.34 Yes: 83
Median : 60.62
Mean : 60.39
3rd Qu.:105.54
Max. :107.54
```

knn implementation in R



So, when we execute this command summary of service train, you will get the details about all the numeric variables which are 5 point summaries including mean and for the service variable which is the categorical variable it gives how many no's are there in that particular attribute and how many yes values are there in that particular attribute.

(Refer Slide Time: 17:45)

Data Science for Engineers

## Summary of ServiceTest

```
> summary(ServiceTest)
 OilQual EnginePerf
Min. : 2.597 Min. : 1.891
1st Qu.: 26.696 1st Qu.: 27.418
Median : 61.023 Median : 61.501
Mean : 58.629 Mean : 59.077
3rd Qu.:104.229 3rd Qu.:103.744
Max. :106.389 Max. :105.744
 NormMileage TyreWear
Min. : 3.589 Min. : 6.143
1st Qu.: 31.260 1st Qu.: 28.901
Median : 59.351 Median : 61.304
Mean : 59.118 Mean : 60.864
3rd Qu.:103.051 3rd Qu.:106.173
Max. :105.051 Max. :108.173
 HVACwear Service
Min. : -1.72 No :99
1st Qu.: 31.31 Yes:36
Median : 62.62
Mean : 58.99
3rd Qu.:105.33
Max. :105.83
```

knn implementation in R



You can use the same summary on service test and you can see that it will return you the 5 point summary for all the numeric variables and

it will return you the number of no values and yes values in the service test.

Let us keep this number in mind we have 99 no values and 36 yes values in the service test. As I said earlier, we are going to act in such a way that we do not know the true yes and no values and we use knn to predict which of them are yes and which of them are no.

(Refer Slide Time: 18:20)

Data Science for Engineers

## Implementation of k-nearest neighbours:`knn()`

```
knn(train, test, cl, k = 1)
```

**Arguments**

|       |                                                                                                         |
|-------|---------------------------------------------------------------------------------------------------------|
| train | matrix or data frame of training set cases.                                                             |
| test  | matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case. |
| cl    | factor of true classifications of training set                                                          |
| k     | number of neighbours considered.                                                                        |

knn implementation in R



Now, let us do the important task as far as this lecture is concerned which is implementation of K-nearest neighbours in R. As I said earlier, the function which we use to implement this K-nearest neighbours is knn function. This knn function takes several arguments but I have listed few which are very important as far as this course is concerned. The arguments it takes are train, test, cl and k.

Let us see what each of this mean. Train is essentially a matrix or a data frame of the training set cases. That means you need to give all the data. In this case this is our service train data frame. And this test is a matrix or data frame for the test set cases. In this case, what will be our test matrix or a data frame this will be our service test data frame. This cl is a factor of true classifications of a training set and this k is the important parameter which is the number of neighbours that are needed to be considered while you do this algorithm which works on this majority voting criteria.

Now, let us implement this knn on our data. How do you do that? So, the way you do it is as follows.

(Refer Slide Time: 19:51)

Data Science for Engineers

## Applying knn algorithm on data

```
Applying K-NN algorithm
K Nearest neighbour is a lazy algorithm and can do prediction directly with the testing
dataset, command "knn", accepts training and testing datasets the class variable of interest
i.e outcome categorical variable is provided for the parameter "cl". parameter "k" is to
specify the number of nearest neighbours required.

predictedknn <- knn(train = ServiceTrain[,-6],
 test = ServiceTest[,-6],
 cl = ServiceTrain$Service,
 k = 3)
```

- `ServiceTrain[,-6]` gives information in `ServiceTrain` except the last column
- `ServiceTest[,-6]` gives information in `ServiceTest` except the last column
- `ServiceTrain$Service` gives the last column of training data as a classification factor to the algorithm

knn implementation in R



There are certain comments here, let us study what those comments are. So, as we have seen in the previous lecture, K-nearest neighbour is a lazy algorithm and can do prediction directly with the testing data set. It accepts training and testing data sets and the class variable of interest that is outcome categorical variable and the parameter `k` as I have mentioned is to specify the number of nearest neighbours that are to be considered for the classification.

So, the way I implement this knn algorithm is through this `knn` command. As a training data set I will give all my service train dataset. Remember I have a negative 6 here, I will talk about it while later. And the test data set what I have given is the attributes in the service test except the 6th column. And in the class variables, I have given this 6th column as my classification parameter.

And let us say I want to build a knn which takes the number of nearest neighbours as 3. So, these are the input arguments for this `knn` function. When I execute this whole command here, it will calculate the labels for the test data set and store them in this predicted knn. I will show you the results in the coming slide.

Meanwhile let us interpret the service train a square bracket and - 6 means this if you remember since service train is a data frame from a data frames lectures. The statement here means that in the service train data frame take all the rows and exclude column 6 that is what it says. This command here gives information in service train except the last column. Similarly, this command here gives the information in the service test except the last column and service train dollar symbol service gives the last column of the training data as a classification factor for the algorithm.

Once you give all these parameters, execute this. The knn will classify the test data points and then store the labels in this predicted knn. Let us look at the results and what this predicted knn contains.  
(Refer Slide Time: 22:32)

Data Science for Engineers

## Results: predicted classes

- “predictedknn” is the output from the algorithm, which has a categorical variable “Yes” or “No”, indicating whether service is needed or not for each case in Test data

```
> # printing the information in predictedknn
> predictedknn
[1] No No No No No No Yes Yes No No
[12] No No No No Yes No No Yes Yes No
[23] Yes No No No No No No No No No
[34] No No Yes No No No No Yes No Yes
[45] No No No Yes Yes No Yes No Yes No
[56] No No No No No No No Yes Yes
[67] Yes No Yes No Yes No No No No
[78] No Yes Yes Yes Yes No Yes No No Yes Yes
[89] Yes No No No Yes No Yes No No No
[100] No No No No No Yes No No No No
[111] No Yes No Yes No Yes Yes No Yes No No
[122] No No No Yes No No No No No No
[133] Yes No No
Levels: No Yes
```

knn implementation in R



So, as we have seen in the earlier slide, predicted knn is the output from the algorithm, which has categorical variable yes or no indicating whether service is needed or not for each case in the test data. When you print this predicted knn, this is the output you see. It essentially says in this 135 values you have, first car no service is needed, and second car no service is needed, and for the 23rd car service is needed and so on.

So, that is what this knn algorithm does and you have actually finished your job of classifying the test cars as whether the service is needed or not. When you do not have this luxury of knowing the true value this is where you stop. But in R case what happened is, we already have the true values whether service is needed or not for this data set what we have. Now, when you have this luxury of knowing the true classes, you can generate what is called confusion matrix and see how well you have classified this performing.

(Refer Slide Time: 23:51)

Data Science for Engineers

## Results: generating confusion matrix manually

```
Command to develop and print a confusion matrix
conf_matrix = table(predictedknn,ServiceTest[,6])

predictedknn No Yes
 No 99 0
 Yes 0 36

A measure of accuracy is calculated by summing the true
positives and true negatives and dividing them by total
number of samples
knn_accuracy = sum(diag(conf_matrix))/nrow(ServiceTest)

> knn_accuracy
[1] 1
```

knn implementation in R



So, there are two ways of generating this confusion matrix. One you can generate the confusion matrix manually, the other way is to use this caret package which can generate confusion matrix and along with it lot of other parameters what Prof. Raghu has talked about in his performance matrix lecture. Let us see how to generate this confusion matrix manually. So, this predicted knn is the labels that is being predicted using the knn algorithm. And when you observe this command here, this is the last column of the service test data frame which says the true labels of whether the service is needed or not.

When I do the table, it generates contingency table and it stores the result in this confusion matrix. When I print this confusion matrix, the result what I see is as follows. This is the predicted no and yes and these are the true no and yes. Recall that we have seen in your test data service is not needed for 99 cars and service is needed for 36 cars. This knn has exactly predicted all of them correctly, this is what is confusion matrix.

(Refer Slide Time: 26:28)

```
confusionMatrix command shown below used from caret package
COnF_Matrix <-confusionMatrix(data = predictedknn,ServiceTest$Service)

> COnF_Matrix
Confusion Matrix and Statistics

Reference
Prediction No Yes
 No 99 0
 Yes 0 36

Accuracy : 1
95% CI : (0.973, 1)
No Information Rate : 0.7333
P-Value [Acc > NIR] : < 2.2e-16
```

knn implementation in R

What we have seen this is the way you generate the confusion matrix manually. Once you have this confusion matrix, you can calculate the accuracy.

So, how do you calculate the accuracy the formula of accuracy is given in Prof. Raghu's performance matrix lecture. Essentially, I am taking the diagonal elements that is the correctly predicted values divided by the total number of entries in the service test. When you divide that you will get the accuracy as  $99 + 36 = 135$ , and the n row of service is also 135. This command here diag of confusion matrix take this element 99 and 36 and the some command will sum them up. And when you divide that with the number of rows in the service test that is 135 by 135, you will get the value of knn accuracy as 1.

Since, knn is managed to predict all the no cases has correctly has no and all the yes cases correctly as yes, your accuracy is 1. This is how you generate the confusion matrix manually. Now, let us see how to generate this confusion matrix using the caret package and the command confusion matrix.

So, the command to generate confusion matrix, which is there in this caret package, is confusion matrix. And the input arguments that you need to give are the predicted labels and the true labels. When you pass these two arguments, this is the confusion matrix that is generated along with confusion matrix it will generate whole lot of other parameters. We have already calculated accuracy manually. We have seen that that is 1. You can also compare now the confusion matrix functions also giving this accuracy as 1.

(Refer Slide Time: 27:12)

Along with this confusion matrix, we will also get a lot of parameters such as sensitivity, specificity, etcetera. So, the reason why you have sensitivity = 1. And specificity = 1 in this case is because all the positive classes are correctly classified all the negative classes are also correctly classified that is the reason why you have the ideal values of one and one for sensitivity and specificity.

So, the balance accuracy is again sensitivity + specificity by 2 which is 2 by 2, it is 1. So, this is how one can implement this knn algorithm in R.

(Refer Slide Time: 27:46)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data Science for Engineers'. Below the header is a light grey section with the title 'Conclusion' in bold black font. Underneath the title is a bulleted list of six items, each starting with a green dot and followed by text in black. The items are:

- `read.csv()` can be used to read data from .csv files
- `str()` function gives data types of each attribute in the given R-object
- `summary()` provides a summary of R-objects
- K-nearest neighbors is supervised learning technique –needs labelled data
- In R knn algorithm can be implemented using `knn()`

At the bottom of the slide is a dark blue footer bar with small white icons and the text 'knn implementation in R'. To the right of the slide is a video player window showing a man speaking against a green background. The video player has a dark blue header bar with small white icons.

In summary what we have seen in this lecture is how to read the dot csv files, how to use the structure and summary functions to know the data types and the summary of R objects and how to implement this K-nearest neighbours algorithm, which is a supervised learning algorithm which needs labelled data. And we have also seen how to implement this K-nearest neighbours algorithm in R using this knn function.

So, with this we end this tutorial session on how to implement knn algorithm in R. In the next lecture, Prof. Raghu will talk about this k means clustering algorithm after which I will come back with a case study on how to implement k means clustering.

Thank you.

**Data science for Engineers**  
**Prof. Raghunathan Rengaswamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 48**  
**K-means Clustering**

So, we are into the last theory lecture of this course. I am going to talk about K-means clustering today. Following this lecture there will be a demonstration of this technique on an case study by the TAs of this course. So, what is K-means clustering? So, K-means clustering is a technique that you can use to cluster or partition a certain number of observations, let us say  $N$  observations, into  $K$  clusters.

(Refer Slide Time: 00:54)

**Data science for Engineers**

## What is K-means clustering?

- A technique to partition  $N$  observations into  $K$  clusters ( $K \leq N$ ) in which each observation belongs to cluster with nearest mean
- One of the simplest unsupervised algorithms
- Works well for all distance metrics where mean is defined (ex. Euclidean distance)

**K-means clustering**

The number of clusters which is  $K$  is something that you either choose or you can run the algorithm for several case and find what is an optimum number of clusters that this data should be partitioned into.

Just a little bit of semantics. I am teaching clustering here under the heading of classification. In general, typical classification algorithms that we see are usually supervised algorithms, in the sense that the data is partitioned into different classes and these labels are generally given.

So, these are labelled data points and the classification algorithms job is to actually find decision boundaries between these different

classes. So, those are supervised algorithms. So, an example of that would be the K nearest neighbour that we saw before. Where we have labelled data and when you get a test data, you kind of bin it into the most likely class that it belongs to. However, many of the clustering algorithms are unsupervised algorithms in the sense that you have N observations as we mentioned here but they are not really labelled into different classes.

So, you might think of clustering as slightly different from classification, where you are actually just finding out if there are data points which share some common characteristics or attributes. However, as far as this course is concerned, we would still think of this as some form of classification or categorization of data into groups. Just that we do not know how many groups are there a priori. However, for a clustering technique to be useful, once you partition this data into different groups as a second step, one would like to look at whether there are certain characteristics that kind of pop out from each of this group to understand and label these individual groups in some way or the other.

So, in that sense we would still think of this as some form of categorization which I could also call as classification into different groups without any supervision. So, having said all of that K-means is one of the simplest unsupervised algorithms where, if you give the number of clusters or number of categories that exist in the data then, this algorithm will partition these N observations into these categories. Now, this algorithm works very well for all distance matrix you again need a distance metric as we will see where we can clearly define a mean for the samples.

(Refer Slide Time: 04:32)

Data science for Engineers

## Description of K-means clustering

Given  $N$  observations  $(x_1, x_2, \dots, x_N)$ , K-means clustering will partition  $n$  observations into  $K$  ( $K \leq N$ ) sets  $S = \{S_1, \dots, S_K\}$  so as to minimize the within cluster sum of squares (WCSS)

$$\arg \min_S \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

Where  $\mu_i$  is the mean of points in  $S_i$

So, when we were talking about optimization for data science, I told you that you know all kinds of algorithms that you come up with in machine learning there will be some optimization basis for these algorithms. So, let us start by describing what K-means clustering optimizes or what is the objective that is driving the K-means clustering algorithm. So, as we described in the previous slide there are  $N$  observations  $x_1$  to  $x_N$  and we are asking the algorithm to partition this into  $K$  clusters. So, what does it mean when we say we partition it into  $K$  clusters? So, we will generate  $K$  sets  $s_1$  to  $s_k$  and we will say this data belongs to this set and this data belongs to the other set and so on.

So, to give a very simple example, let us say you have this observations  $x_1$ ,  $x_2$  all the way up to  $x_N$  and just for the sake of argument let us take that we are going to partition this data into two clusters ok. So, there is going to be one set for one cluster, cluster 1 and there is going to be another set for the other cluster 2. Now the job of K-means clustering algorithm would be to put these data points into these two bins. So, let us say this could go here this could go here,  $x_3$  could go here  $x_N$  could go here maybe an  $x_4$  could go here and so on.

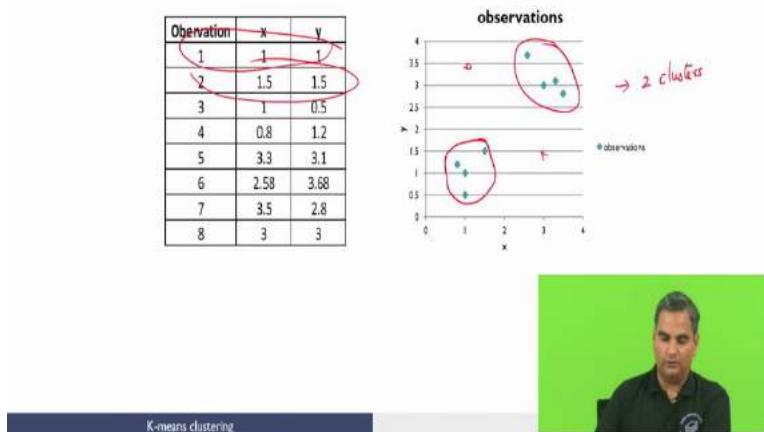
So, all you are doing is you are taking all of these data points and putting them into two bins and while you do it what you are looking for really in a clustering algorithm is to make sure that all the data points that are in this bin have certain characteristics which are like, in the sense that, if I take two data points here and two data points here. These two data points will be more like and these two data points will be more like each other, but if I take a data point here and here they will be in some sense unlike each other. So, if you cluster the data this way then it is something that we can use where we could say look all of these data points share certain common characteristics and then we are going to make some judgments based on what those characteristics are.

So, what we really would like to do is the following. We would like to keep these as compact as possible. So, that like data points are grouped together which would translate to minimizing within cluster, sum of square distances. So, I want this to be a compact group. So, mathematical way of doing this would be the following. So, if I have  $K$  sets into which I am putting all of this in. You take set by set and then make sure that if you calculate a mean for all of this data, the difference between the data and the mean square in a norm sense is as minimum as possible for each cluster and if you have  $K$  clusters you kind of sum all of them together and then say I want a minimum for this whole function. So, this is a basic idea of K-means clustering.

So, there are, there is a double summation. The first summation is for all the data that has been identified to belong to a set. I will

calculate the mean of that set and I will find out a solution for this these means in such a way that this within cluster distance  $x$  which is in the set - I mean is minimized not only for one cluster, but for all clusters. So, this is how you will define this objective. So, this is the objective function that is being optimized and as we have been mentioned mentioning before this  $\mu_i$  is a mean of all the points in set  $s_i$ .

(Refer Slide Time: 09:10)



Now, let us look at how an algorithm such as K-means works. It is a very simple idea. So, let us again illustrate this with a very simple example. Let us say there are two dimensions that we are interested in  $x$  and  $y$  and there are eight observations.

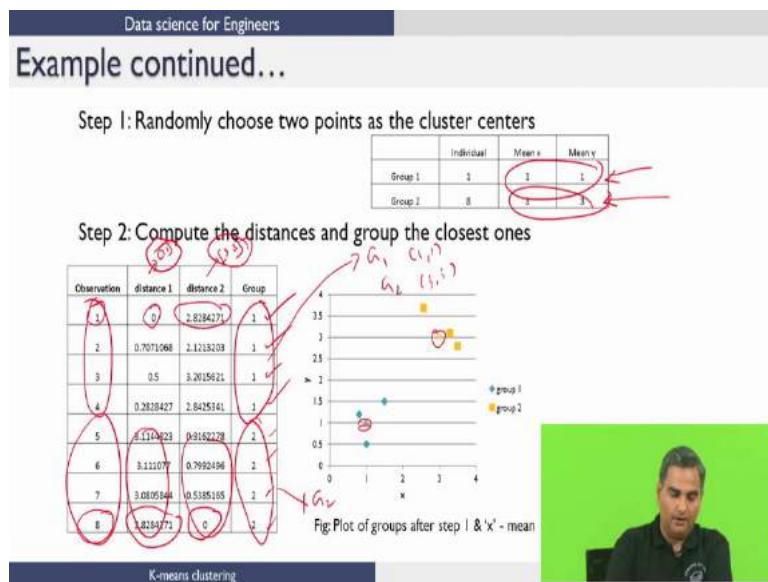
So, this is one data point this is another data points on. So, there are 8 data points and if you simply plot this you would see this and when we talk about cluster. So, notionally you would think that there is a very clear separation of this data into two clusters.

Now, again as I mentioned before in one of the earlier lectures on data science, very easy to see this in two dimensions, but the minute we increase the number of dimensions and the minute we increase the complexity of the organization of the data which you will see at the end of this lecture itself, you will find that finding and the number of clusters is really not that obvious. In any case let us say we want to identify or partition this data into different clusters and let us assume for the sake of illustration with this example that we are interested in partitioning this data into two clusters. So, we are interested in partitioning this data into two clusters.

So, right at the beginning we have no information. So, we do not know that this all belong to one cluster, all of these data points belong to another cluster we are just saying that are going to be two clusters

that is it. So, how do we find the two clusters? Because we have no information labels for these and this is an unsupervised algorithm what we do is we know ultimately there are going to be two clusters. So, what we are going to do is we are going to start two cluster centres in some random location. So, that is the first thing that we are going to do. So, you could start off two clusters somewhere here and here or you could actually pick some points in the data itself to start these two clusters.

(Refer Slide Time: 11:44)



So, for example, let us say we randomly choose two points as cluster centres. So, let us say one point that we have chosen is 1 1, so which would be this point here. And let us assume this is what is group 1 and let us assume that for group 2, we choose point 3 3 which is here. The way we have chosen this, if you go back to the previous slide and look at this, the two cluster centres have been picked from the data itself. So, the one group was observation one the other group groups centre was observation 8.

Now, you could do this or like I mentioned before you could pick a point which is not there in the data also and we will see the impact of choosing this cluster centres later in this lecture.

Now, that we have two cluster centres then what this algorithm does is the following. It finds for every data point in our database, this algorithm first finds out the distance of that data point from each one of this cluster centres. So, in this table we have distance one, which is the distance from the point 1 1 and we have distance two, which is the distance from the point 3 3. Now, if you notice since the first point from the data itself is 1 1 the distance of 1 1 from 1 1 is 0. So, you see

that distance one is 0 and distance two is the distance of the point 1 1 from 3 3.

Similarly since we chose 3 3 as representative of group 2, if you look at point eight which was a 3 3 point the distance of 3 3 from 3 3 is 0 and this is the distance of 3 3 from 1 1 which will be the same as this. For every other point, since they are not either 1 1 or 3 3, there will be distances you can calculate. So, for example, this is a distance of the second point from 1 1 and this is a distance of the second point from 3 3 and if you go back to the previous slide, the second point is the second point is actually 1.5, 1.5 and so on. So, you will go through here each of these points are different from 1 1, 3 3 you will generate these distances. So, for every point you will generate two distance, one from 1 1 other one from 3 3.

Now, since we want all the points that are like 1 1 to be in one group and all the points which are like 3 3 to be in the other group, we use a distance as a metric for likeness. So, if a point is closer to 1 1 then it is more like 1 1 than 3 3 and similarly if a point is close to 3 3 it is more like 3 3 than 1 1. So, we are using a very very simple logic here.

So, you compare these two distances and whichever is a smaller distance you assign that point to that group. So, here distance 1 is less than distance 2. So, this is assigned to group 1 this observation, which is basically the point 1 1. The second observation again if you look at it distance 1 is less than distance 2. So, the second observation is also assigned to group 1 and you will notice through this process a third and fourth will be assigned to group 1 and 5 6 7 8 will be assigned to group 2 because these distances are less than these distances.

So, by starting at some random initial point, we have been able to group these data points into two different groups, one group which is characterized by all these data points the other group which is characterized by these data points. Now, just as a quick note whether this is in two dimensions or N dimensions it really does not matter because the distance formula simply translates and you could have done that for two classes in N dimensions as easily. So, while visualizing data in N dimensions hard this calculation whether it is two or N dimension, it is actually just the same.

Now, what you do is, you know that these group positions are the centres of these groups were randomly chosen now, but we have now more information to update the centres because I know all of these data points belong to group 1 and all of these data points belong to group 2. So, a better representation for this group, so initially the representation for this group was 1 1, a better representation for this group would be the mean of all of these 4 samples and the initial representation for

group 2 was 3.3, but a better representation for group 2 would be the mean of all of these points. So, that is step 3.

(Refer Slide Time: 17:24)

Data science for Engineers

## Example continued...

Step 3: Compute the new mean and repeat step 2

|         | Individual | Means | Mean  |
|---------|------------|-------|-------|
| Group 1 | 1,2,3,4    | 1.075 | 1.05  |
| Group 2 | 5,6,7,8    | 3.095 | 3.145 |

Step 4: If change in mean is negligible or no reassignment then stop the process

Detailed description: This slide shows a K-means clustering example. It includes a table of observations and their assigned groups, a scatter plot of the data points with group centroids, and handwritten annotations explaining the iterative process.

| Observation | distance 1 | distance 2 | Group |
|-------------|------------|------------|-------|
| 1           | 0.9901588  | 2.9903412  | 1     |
| 2           | 0.189709   | 2.2812965  | 1     |
| 3           | 0.5550901  | 3.374174   | 1     |
| 4           | 0.3132481  | 3.0083301  | 1     |
| 5           | 3.0254132  | 0.2098809  | 2     |
| 6           | 3.0301691  | 0.7425966  | 2     |
| 7           | 2.9805038  | 0.9320144  | 2     |
| 8           | 2.7400958  | 0.1733494  | 2     |

K-means clustering

So, we compute the new mean and because group 1 has points 1, 2, 3, 4, we do a mean of those points and the x is 1.075 and y is 1.05. Similarly for group 2, we do the mean of the labels are the data points 5, 6, 7, 8 and you will see the mean here. So, notice how from 1 1 and 3 3 the mean has been updated. In this case because we chose a very simple example the updation is only slight. So, these points move a little bit.

Now, you redo the same computation because I still have the same eight observations but now group 1 is represented not by 1 1, but by 1.075 and 1.05 and group 2 is represented by 3.95 and 3.145 and not 3 3. So, for each of these points you can again calculate a distance 1 and distance 2, and notice previously this distance was 0 because the representative point was 1 1. Now that the representative point has become 1.075 and 1.05 this distance is no more 0, but it is still a small number. So, for each of these data points with these new means you calculate these distances. And again use the same logic to see whether distance 1 is smaller or distance 2 smaller and depending on that you assign these groups.

Now, if we notice that by doing this there was no assignment reassignment that happened. So, whatever were the samples that were originally in group 1 remained in group 1 and whatever are the samples which are in group 2 re-main in group 2. So, if you again compute a mean because the points have not changed the mean is not going to change. So, even after this the mean will be the same. So, if you keep

repeating this process there is no never going to be any reassignment. So, this clustering procedure stops.

Now, if it were the case that because of this new mean let us say for example, if this had gone into group 2 and let us say this and this had gone into group 1 then correspondingly you have to collect all the points in group 1 and calculate a new mean collect all the points in group 2 and calculate a new mean. And then do this process again and you keep doing this process till the mean change is negligible or no reassignment. So, one of these two things happen then you can stop the process. So, this is a very very simple technique.

Now, you notice this technique is very very easily implementable it does not matter the dimensionality of the data you could have 20 variables, 30 variables the procedure remains the same. The only thing is we have to specify how many clusters that are there in the data.

And also remember that this is a unsupervised learning. So, we originally do not know any labels, but at the end of this process at least what we have done is, we have categorized this data into classes or groups. Now if you find something specific about this group by further analysis, then you could basically sometimes maybe even be able to label these groups and the broad categories if that is not possible at least you know from a distance viewpoint that these two might be a different behavior from the system viewpoint. So, from an engineering viewpoint why would something like this be important?

If I let us say give you data for several variables temperatures, pressures, concentrations and so on now for several variables then you could run a clustering algorithm purely on this data and then say I find actually that there are two clusters of this data. Then a natural question an engineer might ask is if the process is stable I would expect all of the data points to be like. But since it seems like there are two distinct groups of data either both or normal but there is some reason why there is this two distinct group or maybe one is normal and one is not really normal, then you can actually go on probe of these two groups which is normal which is not normal and so on.

So, in that sense an algorithm like this allows you to work with just raw data without any annotation. What I mean by annotation here is without any more information in terms of labelling of the data and then start making some judgments about how this data might be organized. And you can look at this multi dimensional data and then look at what is the optimum number of clusters, maybe there are 5 groups in this multi dimensional data which would be impossible to find out by just looking at an excel sheet.

But once you do an algorithm like this then it maybe organizes this into 5 or 6 or 7 groups then that allows you to go and probe more into

what these groups might mean in terms of process operations and so on. So, it is an important algorithm in that sense.

(Refer Slide Time: 22:56)

Data science for Engineers

## Determining number of clusters(K)

- Elbow method – looks at percentage of variance explained as a function of number of clusters
- The point where marginal decrease plateaus is an indicator of the optimal number of clusters
- We will see a demonstration of this in the example



K-means clustering

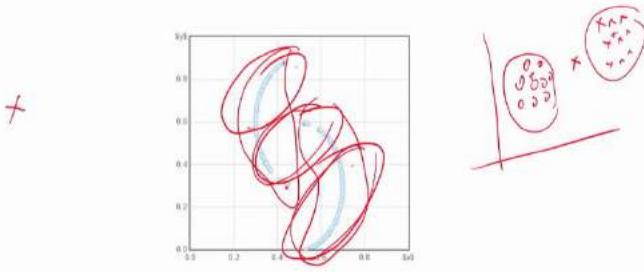
Now, we kept talking about finding the number of clusters. Till now I said you let the algorithm know how many clusters you want to look for, but there is something called an elbow method where you look at the results of the clustering for different number of clusters and use a graph to find out what is an optimum number of clusters. So, this is called an elbow method and you will see a demonstration of this in an example.

(Refer Slide Time: 23:56)

Data science for Engineers

## Disadvantages of K-means

- This algorithm could converge to a local minima, therefore role of initial position is very important
- If the clusters are not spherical, then K-means can fail to identify the correct number of clusters



K-means clustering

The case study that follows this lecture, you will see how this plot looks and how you can make judgments about the optimal number of clusters that you should use. So, basically this approach uses what is called a percentage of variance explained as a function of number of clusters but those are very typical looking plots and you can look at those and then be able to figure out what is the optimal number of clusters.

So, I explained how in an unsupervised fashion you can actually start making sense out of data. This is a very important idea for engineers because this kind of allows a first level categorization of data just purely based on data and then once that is done then one could bring in their domain knowledge to understand these classes and then see whether there are some judgments that one could make.

Couple of disadvantages of K-means that I would like to mention. The algorithm can be quite sensitive to the initial guess that you use. It is very easy to see why this might be. So, let me show you a very simple example. So, let us say I have data like this. So, if my if I start my cluster points initial cluster points here and here you can very clearly see by the algorithm all these data points will be closer to this.

So, all of this will be assigned to this group and all of these data points are closer to this. So, they will be assigned to this group then you will calculate the mean and once a mean is calculated there will never be any reassignment possible afterwards. Then you have clearly these two clusters very well separated.

However, if just to make this point, supposing I start these two cluster centres for example, one here and one somewhere here. Then what is going to happen is that when you calculate the distance between this cluster and all of these data points and this cluster center and all of these data points, it is going to happen that all these data points are going to be closer to this and all of these data points are going to be closer to this than this point.

So, after the first round of the clustering calculations, you will see that the center might not even move because the mean of this and this might be some-where in the center. So, this will never move, but the algorithm will say all of these data points belong to this center and this center will never have any data points for it.

So, this is a very trivial case, but it just still makes the point that if you use the same algorithm depending on how you start your cluster centres, you can get different results. So, you would like to avoid situations like this and these are actually easy situations to avoid, but when you have multi dimensional data and lots of data and which you cannot visualize like I showed you here it turns out it is not really that

obvious to see how you should initially. So, there are ways of solving this problem, but that is something to keep in mind. So, every time you run an algorithm if the initial guesses are different you are likely to get at least minor differences in your results.

And the other important thing to notice, look at how I have been plotting this data. Typically I have been plotting to this data so that you know the clusters are in general spherical. But let us say if I have data like this where you know all of this belongs to one class and all of this belongs to another class. Now, K-means clustering can have difficulty with this kind of data simply because if you look at data within this class, this point and this point are quite far though they are within the same class whereas, this point and this point might be closer than this point in this point.

So, if in an unsupervised fashion if you ask K-means clustering to work on this, depending on where you start and so on, you might get different results. So, for example, if you start with 3 clusters you might in some instances find 3 clusters like this. So, though underneath the data is actually organized differently and if these happen to be two different classes in reality, in an unsupervised fashion when you run the K-means clustering algorithm, you might say all of this belongs to one class, all of this belongs to another class, and all of this belongs to another class and so on so.

So, these kinds of issues could be there. Of course, as I said before there are other clustering algorithms which will quite easily handle data that is organized like this but if you were thinking about K-means then these are some of the things to think about.

So, with this we come to the conclusion of the theory part of this course on data science for engineers. There will be one more lecture which will demonstrate the use of K-means on a case study. With that all the material that we intended to cover would have been covered. I hope this was an useful course for you.

We tried to pick these data science techniques so that you get a good flavour of another things that you think about and also actually techniques that you can use for some problems right away. But more importantly our hope has been that this has increased your interest in this field of data science and as you can see that there are several fascinating aspects to think about when one thinks about machine learning algorithms and so on.

And this course would have hopefully given you a good feel for the kind of thinking and aptitude that you might need to follow up on data science. And also the mathematical foundations that are going to be quite important as you try to learn more advanced and complicated

machine learning techniques either on your own or through courses such as this that are available.

Thanks again.

**Data science for Engineers**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 49**  
**K-means implementation in R**

Welcome to this lecture on Implementation of K-means Algorithm in R. In the previous lectures Professor Raghu would have given a brief introduction about this K-means clustering algorithm and the mathematical details of how this K-means algorithm works.

In this lecture what we are going to do is to introduce you to a case study which we use as a means to explain how K-means algorithm can be implemented in R.

(Refer Slide Time: 00:34)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header is a light gray section titled 'In this lecture' in bold. To the right of the title is a white area containing a bulleted list of topics:

- Case study
  - Problem statement
- Solve the case study using R
  - Read the data from a ".csv" file
  - Understand the data
  - k-means() function
  - Interpret the results

At the bottom of the slide is a dark blue footer bar with the text 'K-means implementation in R.' To the right of the footer is a video player window showing a man in a blue shirt speaking against a green background.

First will start with the problem statement of the case study followed by how to solve the case study using R. As a part of the solution methodology we will also introduce the following aspects such as how to read the data from dot csv file, how to understand the already read data which is in the workspace of your R, details about this K-means function and how to interpret the result that are given by this K-means function. Let us first look at the case study.

(Refer Slide Time: 01:30)

Clustering of trips: a case study



K-means implementation in R

We have main this case study as clustering of trips, the reason will become clear when you see the problem statement.

(Refer Slide Time: 01:41)

Data science for Engineers

## Clustering of trips: Problem statement

An Uber cab driver has attended 91 Trips in a week (5 days). He has a facility which continuously monitors the following parameters for each trip  
Trip length, Max speed, Most frequent speed, Trip duration, number of times brakes are used, idling time and number times the horn is being honked.  
Uber wants to group the trips in to certain number of categories based on the details collected during the trip for some business plan. They have consulted Mr. Sam, a data scientist to perform this job and the details of trips are shared in a ".csv" format file with name "tripDetails.csv"



K-means implementation in R

Let us look at the problem statement of the case study. An Uber cab driver has attended 91 trips in a week. He has a facility in the car which continuously monitors the following parameters for each trip such as trip length, maximum speed, most frequent speed, trip duration, number of times the brakes are used, idling time and number of times the horn is being honked. Uber wants to group this trips into certain number of categories based on the details collected during the trips for some business plan. They have consulted Mister Sam, a data scientist, to perform this job and the details of the trips are shared to him in a dot

csv format file with the name trip details dot csv. This is a problem statement let us look how to solve this case study using R.

(Refer Slide Time: 02:35)

Solution to case study using R

K-means implementation in R

So, to solve this case study first we need to set up our R studio work space.

(Refer Slide Time: 02:41)

Getting things ready

- Setting working directory, clearing variables in the workspace

```
k-means clustering
Set the working directory as the directory
which contains the data files
setwd("Path of the directory with data files")
rm(list=ls()) # to clear the environment
```

K-means implementation in R

You need to copy the file which we have shared with you into the working directory and clear the variables in the workspace. You can set the working directory using the set working directory command and you can pass the path of the directory which contains this data file as

an argument to the set working directory function. Or you can use the GUI as we have specified in the R module to set the working directory. And this command here removes all the variables that are in the workspace and clear the R environment. You can very well use the brush button to clear all the variables from the environment.

(Refer Slide Time: 03:28)

Data science for Engineers

## Reading the data

- Data for this case study is provided to you file with name “tripDetails.csv”
- To read the data from a “.csv” file we use `read.csv()` function



K-means implementation in R

The next step is to read the data from the given file. And data for this case study is provided in a file with name trip details dot csv. If you notice the extension of this file is dot csv which means comma separated value file. In R to read the data from a dot csv file we use `read.csv` function.

(Refer Slide Time: 04:01)

Data science for Engineers

## `read.csv()`

Reads a file in table format and creates a data frame from it

SYNTAX

```
read.csv(file, row.names=1)
```

|                        |                                                                                                                                                                                                                                            |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>file</code>      | the name of the file which the data are to be read from. Each row of the table appears as one line of the file.                                                                                                                            |
| <code>row.names</code> | a vector of row names. This can be a vector giving the actual row names, or a single number giving the column of the table which contains the row names, or character string giving the name of the table column containing the row names. |



K-means implementation in R

Let us look what does the read.csv function takes as input argument and what it gives us an output. read.csv reads a file in the table format and creates a data frame from it. The syntax of the read.csv is as follows; read dot csv it takes two arguments the first argument is a file name and the second argument is the row names we will see what this individual arguments are about.

The file is the name of the file from which the data has to be read, row names is a vector of row names. This can be a vector giving the actual row names are a single number specifying which column of the table contains this row names. So, essentially the syntax is read dot csv, the filename and if you have a column which specifies the row names you can give that particular column as row names. In this case we have the first column as the row indices that is the reason why we have given this row dot names as 1.

(Refer Slide Time: 05:08)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. Below the header, the main title 'Reading the data' is displayed in a large, bold, black font. To the right of the title, there is a small video window showing a man with dark hair and a blue shirt against a green background. The main content area contains a bulleted list and a code snippet. The list says: '• Data for this case study is provided to you file with name "tripDetails.csv"'. Below the list is a code block in a monospaced font:

```
#Reading the data
tripDetails = read.csv("tripDetails.csv",
 row.names=1)
```

Let us see how to read the data from the trip details dot csv. The data from this tripdetails.csv can be read by executing this following command here. I am using read dot csv command and I am trying to read the data from tripdetails.csv and I know that in the first column of the csv file I have the row names. Therefore, I have specified row dot names as one and this is the filename from which I want to have read the data.

And assigning the data which is read from this read dot csv to the object called trip details. As mentioned in the help of read.csv, it reads the data from this tabular format and then assign it to object call trip details which is of type data frame.

(Refer Slide Time: 06:07)

The screenshot shows the RStudio interface with the title bar "Data science for Engineers". A green box highlights the command "View(tripDetails)". Below it, the R console window displays the command "View(tripDetails)". The main workspace shows a data frame named "tripDetails" with 91 rows and 7 columns. The columns are labeled: TripLength, MaxSpeed, MostFreqSpeed, TripDuration, Brakes, IdlingTime, and Honking. The data frame contains numerical values for each trip. A small video player in the bottom right corner shows a person speaking.

Once you get this data onto your R environment you can view the data frame using view function. Notice this is capital V and once you run this command it will pop up a tabular column in your editor window which shows the variables in the data frame and the number of entries in the data frame.

In this case we have 7 variables and around 91 entries. This is how you can view the data frame once it is loaded into your workspace.

(Refer Slide Time: 06:50)

The screenshot shows the RStudio interface with the title bar "Data science for Engineers". A green box highlights the word "Variables". To the right, a callout box provides a detailed description of the data frame: "Data contains 91 Trips where 7 variables (columns) named Trip length, Max speed, Most Freq. speed, Trip duration, Brakes, Idling time and Honking are noted for each trip". Below this, another callout box indicates "91 observations". A small video player in the bottom right corner shows a person speaking.

So, to make it very clear, we have this 7 variables and there are 91 observations in this data frame. And the 7 variables are trip length,

maximum speed, most frequent speed, trip duration, brakes, idling time and honking are noted for each trip.

(Refer Slide Time: 07:13)

Data science for Engineers

## Structure of the data

- Structure of data
  - Variables and their data types
- **str()**

Compactly display the internal structure of an R object

SYNTAX

```
str(object)
```

object      any R object about which you want to have some information.

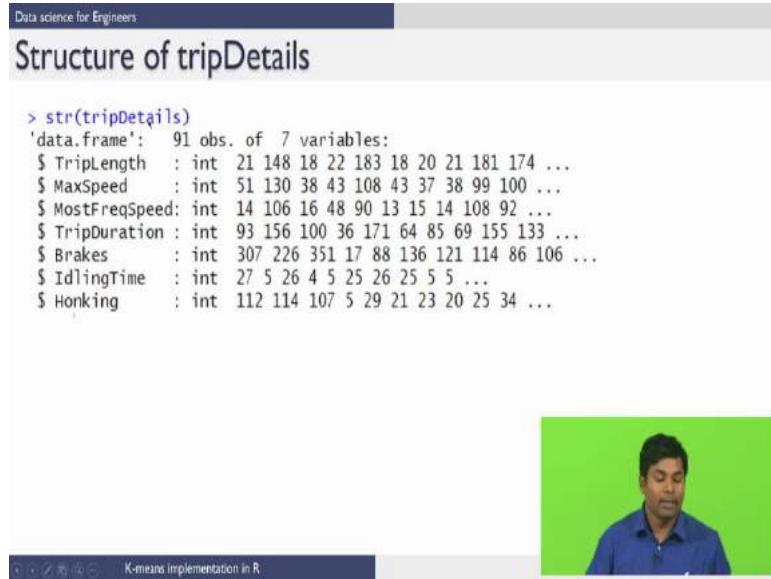
K-means implementation in R



Now, that we have seen how a data frame looks, it is time to see what are the data types of each variable that is available in the data frame. What is the way one can do that? One has to use the structure function to do that. The structure function compactly displays the internal structure of an R object.

The syntax of the structure function is as follows, structure and the argument it takes place an R object. This object is an any R object about which you want to have some information.

(Refer Slide Time: 08:00)



```
> str(tripDetails)
'data.frame': 91 obs. of 7 variables:
 $ TripLength : int 21 148 18 22 183 18 20 21 181 174 ...
 $ MaxSpeed : int 51 130 38 43 108 43 37 38 99 100 ...
 $ MostFreqSpeed: int 14 106 16 48 90 13 15 14 108 92 ...
 $ TripDuration : int 93 156 100 36 171 64 85 69 155 133 ...
 $ Brakes : int 307 226 351 17 88 136 121 114 86 106 ...
 $ IdlingTime : int 27 5 26 4 5 25 26 25 5 5 ...
 $ Honking : int 112 114 107 5 29 21 23 20 25 34 ...
```

Now, let us look at the structure of the data frame which we have extracted from the trip details dot csv. The data frame which we have extracted from trip details dot csv is trip details and I am passing that data frame as an argument to this structure function. When I execute this command, it will show that trip details is a data frame which contains 91 observations of 7 variables and the 7 variables are trip length, maximum speed, most frequent speed and so on.

And correspondingly it gives what is the data type of these variables. If you can notice, all of them are integer type data variables. Keep this in mind because the K-means wants all the variables in the data matrix as the numeric variables or integer variables. We will see that when we discuss about the K-means function as we go along.

Now, structure gives you type of the object and variables that are there in the object and their data types.

(Refer Slide Time: 09:12)

Data science for Engineers

## Summary of the data

- Summary of data
  - Five point summary of the numeric variables
- **summary()**

Summary is a generic function used to produce result summaries of the results of various model fitting functions and five point summaries of numeric R objects

SYNTAX

```
summary(object)
```

object any R object about which you want to have some information.



K-means implementation in R

There is another command which is summary which gives you the five point summary of the numeric variables. This is the function summary. Summary function is a generic function used to produce results summaries of the results of various models and five point summaries of numeric R objects. The syntax for the summary function is as follows.

The summary function takes one argument which is an R object. This object is again any R object about which you want to know some information. Let us see the summary for our data frame trip details.

(Refer Slide Time: 09:50)

Data science for Engineers

## Summary of tripDetails

```
> summary(tripDetails)
 TripLength MaxSpeed MostFreqSpeed
Min. : 16.00 Min. : 35.00 Min. : 12.00
1st Qu.: 20.00 1st Qu.: 42.00 1st Qu.: 15.50
Median : 21.00 Median : 54.00 Median : 42.00
Mean : 70.77 Mean : 70.36 Mean : 50.65
3rd Qu.:163.00 3rd Qu.:105.50 3rd Qu.: 89.00
Max. :210.00 Max. :138.00 Max. :118.00
 TripDuration Brakes IdlingTime
Min. : 22.00 Min. : 14.0 Min. : 4.00
1st Qu.: 34.50 1st Qu.: 36.5 1st Qu.: 5.00
Median : 88.00 Median :100.0 Median : 5.00
Mean : 87.37 Mean :135.4 Mean :11.59
3rd Qu.:133.00 3rd Qu.:198.0 3rd Qu.:24.00
Max. :171.00 Max. :429.0 Max. :32.00
 Honking
Min. : 4.00
1st Qu.: 20.00
Median : 25.00
Mean : 49.92
3rd Qu.: 97.50
Max. :155.00
```



K-means implementation in R

When you execute the summary command on your data frame trip details, it will give you five point summary for all the 7 variables of your data frame.

(Refer Slide Time: 10:02)

The screenshot shows a presentation slide with a dark blue header bar containing the text 'Data science for Engineers'. The main title 'K-means clustering' is centered above a bulleted list. The list contains two main points, each with a sub-point. The first point discusses Mr. Sam's job of segregating trips into clusters using k-means clustering. The second point discusses using the 'kmeans()' function in R for k-means clustering on data. At the bottom right of the slide, there is a small video thumbnail showing a person in a blue shirt against a green background. The bottom navigation bar includes icons for back, forward, and search, along with the text 'K-means implementation in R'.

- Given the dataset of trip details, Mr. Sam's job is to segregate these trips into clusters
  - We seek an answer through k-means clustering
- Using k-means clustering on data
  - k-means clustering in R can be applied on data using “`kmeans()`” function

Now, let us look at our primary task of implementing K-means clustering on the data frame. So, we have been given this data set of trip details and we have to segregate the trips into clusters.

We are seeking an answer through the K-means clustering algorithm. If you would have noticed in the data the trips are not labeled as short trip, long trip and so on. This means the data what we have is an unlabeled data and when one wants to learn from this unlabeled data one has to go for unsupervised learning technique. K-means is one such unsupervised learning technique and this K-means clustering in R can be implemented by using this K-means function.

Let us look at what this K-means function takes as input arguments and what does it return.

(Refer Slide Time: 11:03)

Data science for Engineers

## kmeans()

```
object = kmeans(x, centers, iter.max = 10, nstart = 1)
```

Arguments

|          |                                                                                                                                                                           |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x        | numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).                            |
| centers  | either the number of clusters, say k, or a set of initial (distinct) cluster centers. If a number, a random set of (distinct) rows in x is chosen as the initial centers. |
| iter.max | the maximum number of iterations allowed.                                                                                                                                 |
| nstart   | if centers is a number, how many random sets should be chosen?                                                                                                            |
| object   | an R object of class "kmeans", typically the result "ob" of ob <- kmeans(..).                                                                                             |

K-means implementation In R



K-means takes several input arguments. I have given few of important arguments here x stands for numeric matrix of data or the objects that can be coerced to a matrix and this centers we can either give the number of clusters you want to generate out of this data or you can give a set of initial cluster centers.

So, if you specify this k as a number let us say 3 clusters, what it does is it will choose a random set of distinct rows in this numeric matrix x as the initial centers. And as you know K-means is an iterative algorithm. You can set a maximum iteration limit using this iter max argument.

The K-means clustering algorithm depends upon the initial cluster centers, this option here nstart will help you to specify how many sets of different cluster centers has to be used to come up with the model. And finally, the output argument is object which returns an R object which is of class K-means that is basically is a result of a function which is as shown here that is exactly what we have here. When you execute this command, it will take the data matrix and it will take number of clusters you want to build from this data and returns you a K-means R object. Let us now implement this K-means algorithm on our data.

(Refer Slide Time: 12:45)

Data science for Engineers

## Implementing K-means

- Clustering data using k-means and seeing the clusters details

```
k-means clustering using kmeans command
tripCluster <- kmeans(tripDetails,3)
```

K-means implementation In R



This is what we are essentially doing. We are clustering our data using the K-means and we can see what are the details that this K-means gives as an output. This command here takes this data frame trip details and this argument here specifies I want to divide the data into 3 clusters. When I execute this command it will divide the data into 3 clusters and it will assign the result as a trip cluster R object which is essentially a list. Let us see what information does this trip cluster has.

(Refer Slide Time: 13:28)

Data science for Engineers

## Results

tripCluster has the following information

```
> tripCluster
K-means clustering with 3 clusters of sizes 46, 15, 30

Cluster means:
 TripLength MaxSpeed MostFreqSpeed TripDuration Brakes
1 19.91304 48.21739 32.82609 50.13043 59.93478
2 20.26667 45.06667 14.46667 88.73333 350.13333
3 174.00000 116.96667 96.06667 143.80000 143.86667

 IdlingTime Honking
1 11.413043 15.60870
2 25.400000 97.73333
3 4.966667 78.63333

Clustering vector:
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
 2 3 2 1 3 1 1 3 3 1 2 1 1 3 1 1 2 1 3
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
 1 2 2 2 1 2 3 3 1 1 1 3 1 3 3 1 2 3 1
43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63
 1 1 3 1 2 3 1 1 1 1 2 3 1 1 3 1 1 1 3
64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84
 3 2 1 1 3 1 1 2 1 1 1 3 3 3 1 1 3 3 1
85 86 87 88 89 90 91
 1 1 2 3 2 3 1


```

K-means implementation In R



Trip cluster has the following information. The first line essentially gives you it has clustered the data into 3 cluster which are of sizes 46, 16 and 30 and if you sum this up, you will end up with your total number of rows in your data frame that is 91 you can verify that. And for each variable it will give the means of the clusters. So, because you wanted to divide the whole data into 3 clusters the first row is the cluster one information where the mean of the trip length is 19.9 and the mean of the maximum speed is 48.21 and so on.

Similarly the lines 2 and 3 represents the information about the cluster. So, what can one infer from this cluster means? You can actually see for example, mean trip length is 19.9, I can actually say that this is of shortest trip and of the mean trip length is 174 I can treat this as a long trips. So, this information can essentially help the people who wanted to do this analysis and then categorize these clusters into meaningful information depending upon the problem they are looking at.

The next means of information that trip cluster contains is, it will say among 91 rows what does each row belong to. For example, look at here the first element that means, the first row belongs to the cluster 2 and the second element belongs to the cluster 3, total means belongs cluster 2 and so on it will identify each row into one of this clusters. And as you know K-means is a hard clustering algorithm that means, each point has to be allotted to any of the clusters and no two clusters contain similar data point otherwise you cannot have a single data point belonging into 2 clusters.

(Refer Slide Time: 15:51)

Data science for Engineers

## Results

tripCluster has the following information

```
within cluster sum of squares by cluster:
[1] 160740.2 25986.8 194647.0
(between_SS / total_SS = 83.3 %)
Available components:
[1] "cluster" "centers" "totss" "withinss"
[5] "tot.withinss" "betweenss" "size" "iter"
[9] "ifault"
*
```



K-means implementation in R

This trip cluster has some more information which is known as within clusters sum of squares. And if you see this contains 3 elements. That means, how much is the variance in each of this clusters is what this within cluster sum of the squares uses. The lesser the variance the good are the clusters and it will also give what are the other components that you can look from the trip clusters.

Essentially when you build a K-means algorithm it will give you the following information. How many clusters it has built and how many data points are there in each of these clusters and what are the means of each of these clusters and how are each points are categorized into 1 of the 3 clusters which you want to build and the other information.

(Refer Slide Time: 16:59)

The screenshot shows a presentation slide with the title "Results: k calculation". Below the title is a code block in R:

```
Method to calculate optimal k
k.max <- 10 # Maximum 10 clusters assumed
wss <- rep(NA, k.max)
nClust <- list()
for (i in 1:k.max){
 driveClasses <- kmeans(tripDetails, i)
 wss[i] <- driveClasses$tot.withinss
 nClust[[i]] <- driveClasses$size
}
plot(1:k.max, wss,
 type="b", pch = 19,
 xlab="Number of clusters K",
 ylab="Total within-clusters sum of squares:Trips")
```

At the bottom of the slide, there is a small video player window showing a man speaking. The video player has a green background and the text "K-means implementation in R" at the bottom.

So, biggest problem with this K-means algorithm is to figure out what number of clusters has to be given. So, there is one method which is called as the elbow method which can help us to calculate the optimal number of clusters K. For that all you need to do is you have to write one for loop which does lot of K-means algorithms and get the metric which is within sum of squares and when you plot this within sum of squares with the number of clusters you will find out after certain number of clusters the decrease in this within sum of squares value is low where you say look this is the optimal number of clusters into which your data has to be divided.

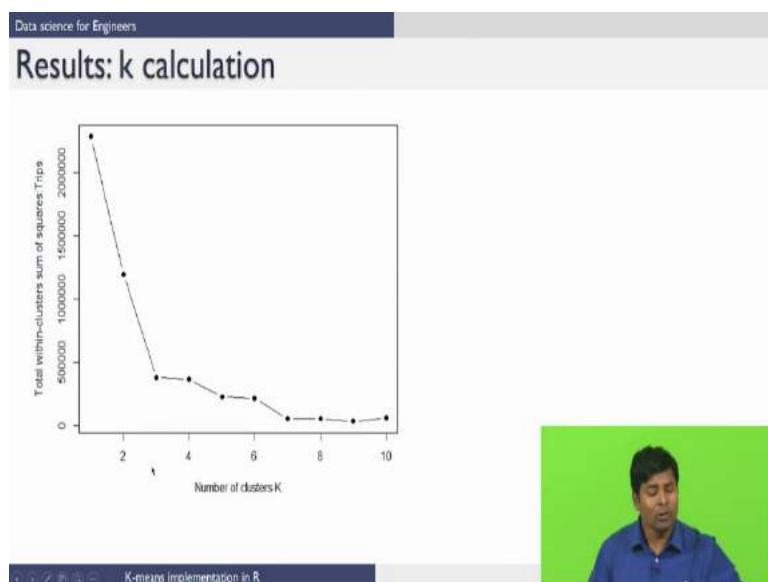
Let us illustrate this code line by line. I am assigning a value of ten to the k max and I am pre allocating a memory which is, so this many number of clusters us there and I am repeating NA's for this many number of times. This is within sum of squares value. Essentially I am

creating a vector with NA's which is of the size 10 by 1. And I am initializing an empty list for the number of clusters.

This for loop will run from 1 to number of maximum clusters that is in this case its 10 and for each value of i this K-means algorithm is implemented, the objects are being stored into this drive classes and in the drive classes the total within sum of squares distance is been allocated into this wss of i. And the size of the cluster is allocated into the components of the list which you have created.

So, once this for loop get executed you will get a vector of the within sum of those errors and a list of the number of clusters. Now, we can plot that with number of clusters in x axis and within sum of squares value in y axis and type = b represents both line and points has to be there and pch = 19 specify the symbol that has to be used along with the plot and x lab and y lab has their normal meaning which are x label and the y label.

(Refer Slide Time: 19:27)



Let us look at the plot. So, this is the x axis which is number of clusters which is varied from 0 to 10 because the maximum number of clusters we had is 0 to 10 and this y axis is total within sum of squares values with respect to trips and if you can see this total within sum of square value drastically decreased when one moved from 1 cluster to 2 clusters and from 2 clusters to 3 clusters and after that decrease in total within sum of squares clusters is not much when compared to the earlier ones.

That is the reason why I can choose this  $K = 3$  as my optimal number of clusters.

(Refer Slide Time: 20:13)

Data science for Engineers

## Summary

- K-means is an unsupervised algorithm
- `kmeans()`
- Elbow method

K-means implementation in R

In summary we have seen that K-means algorithm is an unsupervised learning algorithm. That means, we have the data which is not labeled. In this cases we have to use one of the unsupervised learning algorithms, in this case we have use K-means algorithm and we have seen how to implement this K-means algorithm in R, and we have seen one method to find the optimal number of clusters for K-means which is elbow method.

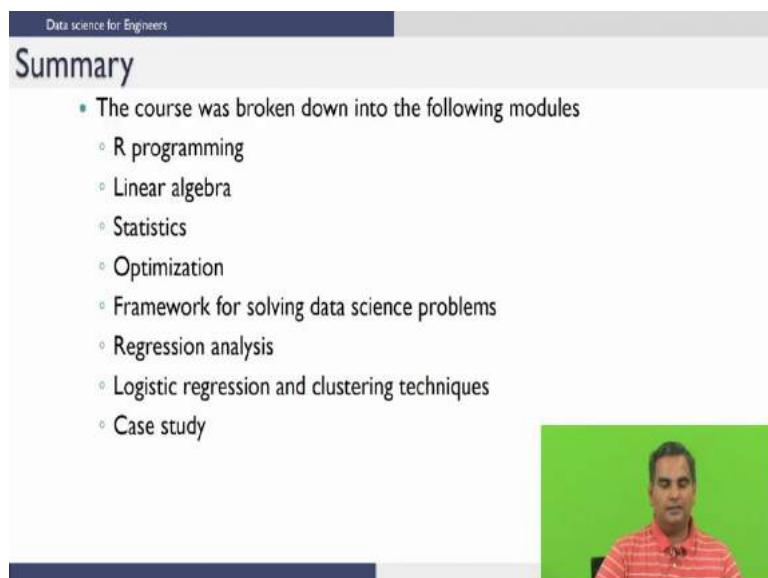
Thank you.

**Data science for Engineers**  
**Prof. Raghunathan Rengaswamy**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Lecture - 50**  
**Data science for engineers – Summary**

So, with this we come to the end of the course on Data Science for Engineers. I am going to do a quick summary of the course. I hope all of you had a productive time looking through the videos and learning interesting ideas in data science.

(Refer Slide Time: 00:34)



The screenshot shows a presentation slide with a dark blue header bar containing the text "Data science for Engineers". Below the header is a light gray section with the word "Summary" in bold black font. Underneath "Summary" is a bulleted list of course modules:

- The course was broken down into the following modules
  - R programming
  - Linear algebra
  - Statistics
  - Optimization
  - Framework for solving data science problems
  - Regression analysis
  - Logistic regression and clustering techniques
  - Case study

At the bottom right of the slide, there is a video player interface showing a thumbnail of a man in a red striped shirt against a green background. The video player has a dark blue progress bar at the bottom.

In summary we broke down this course into the following modules. We started with R programming as the programming platform for solving data science problems. So, we had a set of lectures explaining the important concepts of R programming that are useful from a data science viewpoint.

We followed that up with lectures on fundamental ideas in linear algebra and statistics that are useful for data science and machine learning. We then introduced the idea of optimization and optimization algorithms that are useful for us to understand machine learning algorithms and make sense out of why we get some results when we run certain machine learning algorithms and so on.

After that we described a framework for solving data science problems. This is a framework that you can use to conceptually break down a large data science problems into smaller sub problems in some work ow fashion. We then moved on to describing regression analysis where we described techniques for univariate and multivariate problems.

Here we focused on building appropriate models identifying the goodness of models and so on. Then we looked at classification techniques logistics, regression and kNN and then we also described clustering techniques such as k-means clustering. And as part of this course there is a case study for you to practice which will be available on the website.

(Refer Slide Time: 02:27)

Data science for Engineers

## What next?

- More practice
- More involved data science problems
- More machine learning techniques from an engineers' perspective
  - Decision trees, random forests
  - Support Vector Machines
  - Kernel tricks
  - Spectral clustering and others
  - Structural, probabilistic, constrained PCA, NMF, NCA
  - Deep learning
  - Reinforcement learning

Now, that is you have done this course and hopefully you have learned enough from this course. What is the logical next step if you were excited by this course and want to know more about data science after doing this course? I would say first is to do more practice on the same ideas that I have been taught in this course. So, you might want to look at other problems and other practice examples and exercises for the concepts that we have already taught in this course. So, that is the first thing to do.

Once you do that the data science problems that we described in this course are rather simple. So, you might want to look at how people solve more involved data science problems and whether you can use the framework to break them down into smaller problems and then see whether you can learn more about these problems. Now, as I mentioned before we teach only very few machine learning techniques

in this course for the beginners. However, the many many more algorithms that are out there such as decision trees, random forests, support vector machines, kernel tricks and so on. So, we have listed many of the commonly known and used algorithms for more complicated or more complex machine learning problems.

So, the next logical step would be once you master the material that has been taught in this course is to look at learning these algorithms and you could use the same notion of understanding the assumptions that are underlying these algorithms to get a good idea of why these algorithms work the way they work. And also would understand the technical details of these algorithms in terms of what is the learning rule and why does it work and so on.

So, a data science is a very vast field. Obviously, there is a lot of interest this lot of buzz around this field and it is only going to get even more important as we go along because more data is going to be generated and one would expect the computational power to increase even more for the next few years. And the math and the algorithmic details are being proved by many researchers, so one would expect even better algorithms to come down in the next few years.

So, a mix of all of these factors make it important that everyone have some fundamental understanding of data science and people who are really interested in this field have much more in depth understanding of techniques such as the ones that are listed here and so that you are able to solve much more complicated problems which have much more value in real life. So, happy learning and hope this course was useful.

Thank you.



**THIS BOOK IS NOT FOR SALE  
NOR COMMERCIAL USE**