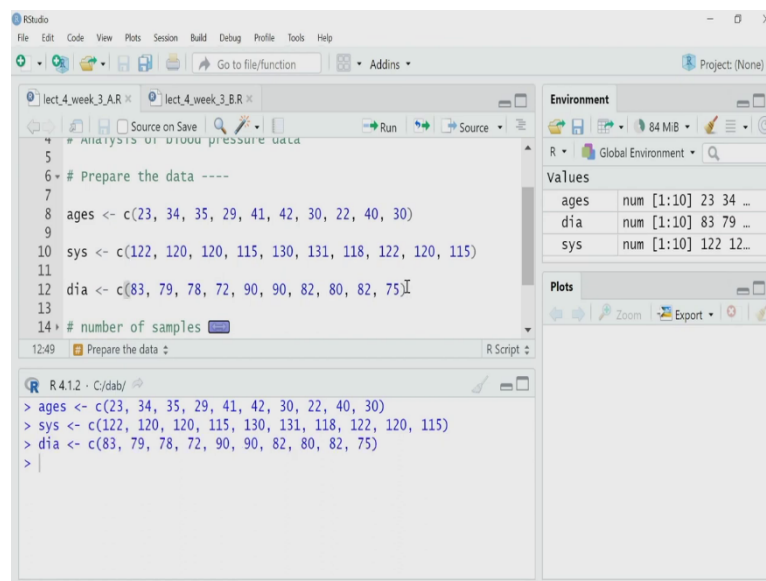


**Data Analysis for Biologists**  
**Professor Biplab Bose**  
**Department of Biosciences and Bioengineering**  
**Mehta Family School of Data Science and Artificial Intelligence**  
**Indian Institute of Technology Guwahati**  
**Lecture 17**  
**Statistics using R – descriptive statistics**

In the last lecture, we learned how to read the data file and then extract the sub part of that data and then write it back to another file. Now, once I get a data set for analysis, I may have to go for detailed statistical analysis or I may have to do some analysis to understand the different features of the data. Before we go for such detailed analysis, most of the time what we do we look into the generalized behavior of the data, generalized properties, overall properties of the data.

That means, we want to look the general statistics of the data, for example, you want to know the mean of different variables, the variances of those, how the data is distributed, like that. So, these are in general called descriptive statistics. So, once you have got the data using R you can actually calculate these descriptive statistics for your data set and then you may make some certain decision to how to analyze the data further. So, in this lecture, I will discuss some of the descriptive statistics that you can perform using R. I will be using R Studio, you can use native R also.

(Refer Slide Time: 01:43)



The screenshot shows the RStudio interface. The main editor window contains the following R code:

```
1 # Analysis of blood pressure data
2
3 # Prepare the data ----
4
5
6 # Prepare the data ----
7
8 ages <- c(23, 34, 35, 29, 41, 42, 30, 22, 40, 30)
9
10 sys <- c(122, 120, 120, 115, 130, 131, 118, 122, 120, 115)
11
12 dia <- c(83, 79, 78, 72, 90, 90, 82, 80, 82, 75)
13
14 # number of samples
```

The Environment pane on the right shows the following variables and their values:

Variable	Class	Values
ages	num [1:10]	23 34 ...
dia	num [1:10]	83 79 ...
sys	num [1:10]	122 12...

The console window at the bottom shows the execution of the code:

```
> ages <- c(23, 34, 35, 29, 41, 42, 30, 22, 40, 30)
> sys <- c(122, 120, 120, 115, 130, 131, 118, 122, 120, 115)
> dia <- c(83, 79, 78, 72, 90, 90, 82, 80, 82, 75)
>
```

To show the utility of R to get descriptive statistics of a data set I will create a data set for a blood pressure experiment. Suppose, I have multiple volunteers and I have measured their

blood pressure. And I know their age and I know their systolic pressure, I know their diastolic pressure. In the last lecture, we have learned how to read a data file, but here what I will do, suppose, I have written the data in a paper and I want to create the data set in R itself.

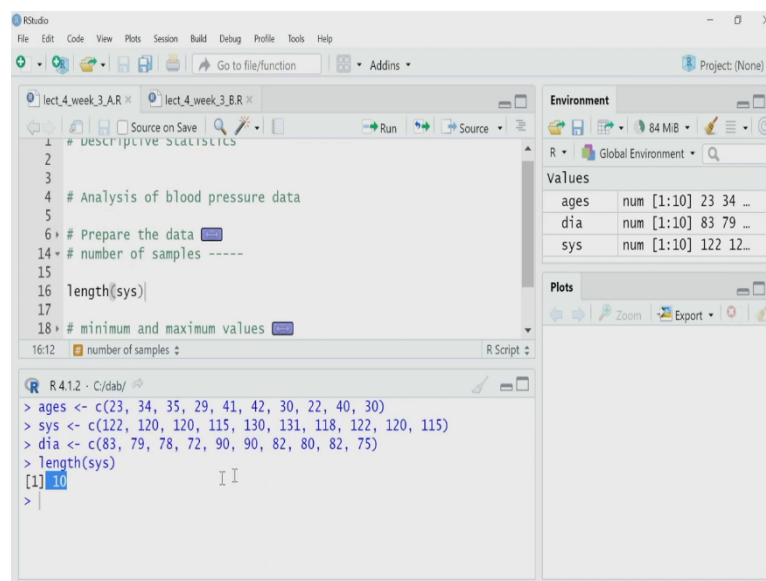
So, let us start doing that. The first thing that I have to do I have to prepare the data. So, I have manually suppose entered that ages, ages is a variable and ages of different person in my data set someone has 23, that another person has the age 34 years, somebody has 42 something like that. So, I use the c function here c(), to join them together to create a vector and I give the name of that vector as ages.

So, let me execute that. So, in the environment pane you can see, ages variable has been created as a numerical variable and it has 10 elements, 10 data points, and those are 23, 34 something like that. Similarly, for systolic blood pressure for each of these individual, I sequentially arrange them and then use c function to create a vector called sys. So again, I am using c function and within the round bracket or the first bracket I am putting the corresponding value.

Remember what I have done this 122, which is the systolic pressure for the person who is the first person in my ages list also, that order I have maintained. And then similarly, diastolic pressure variable, dia is also created using the c function, where I have joined all these diastolic pressure of each of these individual to create a vector. So, let us execute both of these.

So now, I have three variable ages, dia and sys. These are the age, diastolic pressure and systolic pressure for individual and I have these three vectors. And now this is my raw data. Now, on these raw data, I can calculate certain descriptive statistics. So, let us start with something which usually many a time you have to calculate first that what is the size of your data set. In this case, we know that it is there are only 10 people that have 10 samples, but sometime your data set can be very large. So, how do I know how many samples are there in that.

(Refer Slide Time: 04:31)



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 # DESCRIPTIVE STATISTICS
2
3
4 # Analysis of blood pressure data
5
6 # Prepare the data
14 # number of samples -----
15
16 length(sys)
17
18 # minimum and maximum values
```

The console shows the execution of the code:

```
R 4.1.2 · C:/dab/
> ages <- c(23, 34, 35, 29, 41, 42, 30, 22, 40, 30)
> sys <- c(122, 120, 120, 115, 130, 131, 118, 122, 120, 115)
> dia <- c(83, 79, 78, 72, 90, 90, 82, 80, 82, 75)
> length(sys)
[1] 10
>
```

The Environment pane on the right shows the following values:

Variable	Type	Value
ages	num	[1:10] 23 34 ...
dia	num	[1:10] 83 79 ...
sys	num	[1:10] 122 12...

The option to do that is to use something called length function. So, length function, length written here, and in the round bracket, you have to write the name of the variable.

length(sys)

In this case, I have written sys, this is the variable for systolic blood pressure. So, the length will tell me the length of the vector, sys is a vector, it is the list, it has some 10 numbers in that.

So if I use length function to get the length of this variable sys I should get something like this 10 here. So, it has 10 elements in this or 10 data points in the sys variable. Now, once I have got that I may ask, what is the minimum value of systolic pressure, what is the maximum value of systolic pressure, like those questions. So, those are also part of descriptive statistics.

(Refer Slide Time: 05:21)

```

5
6 # Prepare the data
14 # number of samples
18 # minimum and maximum values ----
19
20 min(sys)
21
22 max(sys)
23
24 # Median, mean, variance and std. dev
27 # Create a table that comprehensive statistics
229 minimum and maximum values

```

```

R 4.1.2 · C:/dab/
> sys <- c(122, 120, 120, 115, 130, 131, 118, 122, 120, 115)
> dia <- c(83, 79, 78, 72, 90, 90, 82, 80, 82, 75)
> length(sys)
[1] 10
> min(sys)
[1] 115
> max(sys)
[1] 131
>

```

min(sys)

max(sys)

So, I have to get the minimum systolic pressure of these samples, I have 10 sample and I want to get the minimum systolic pressure out of those, I will use the min function that will give me the minimum number of the variable sys, because I am bothered about the systolic blood pressure right now. So, similarly, if I want to calculate the maximum of these values, I will use the max function and again I am working on the sys variable.

So, here I execute that and R says the minimum value is 115. Quite good enough, because on an average it should be around 120 or something less, less than 120 and the maximum value is 131. Now, I have calculated the minimum and maximum for systolic blood pressure, you can also use this min and max function to get the minimum age of the samples, maximum age of the people in your sample, you can use the same min-max function to get the minimum value for the diastolic pressure as well as the maximum value for the diastolic pressure, I have not shown them but you can easily do that.

Now, few other statistics that are very useful to understand the dispersion of the data and the behavior of the data, you must be knowing is the minimum, median, variance, standard deviation, all these things. And I am sure you know the definition of all those. So, how can I calculate median, mean, variance in some of these suppose systolic data.

(Refer Slide Time: 06:48)

```
18 # minimum and maximum values
24 # Median, mean, variance and std. dev ----
25
26 median(sys)
27
28 mean(sys)
29
30 var(sys)
31
32 sd(sys)
33
```

Environment

Values	
ages	num [1:10] 23 34 ...
dia	num [1:10] 83 79 ...
sys	num [1:10] 122 12...

```
R 4.1.2 · C:/dab/
> median(sys)
[1] 120
> mean(sys)
[1] 121.3
> var(sys)
[1] 29.56667
> sd(sys)
[1] 5.437524
>
```

median(sys)                      mean(sys)                      var(sys)                      sd(sys)

So, here I show that. So, to calculate the median I will use median function as it is shown here median then inside the round bracket I write the name of the variable in this case I have written sys. And then to calculate mean that is the arithmetic mean or the average in colloquial word, it will mean the function is called mean. And again, I am using over the variable sys. So, let me calculate the median and mean.

So, I execute the median function on sys. So it says the median value of systolic blood pressure in my 10 samples, 10 people is 120, whereas the mean is it says 121.3. So they are different. Now, I have calculated the mean. So obviously, I will be tempted to calculate the variance of the systolic blood pressure data. It is very easy to calculate, R has an inbuilt function call var, which will be able to calculate the variance.

And the argument for this function will be obviously the variable for which you are calculating the variance. So, I call var function for sys. And the variance is 29.56, something like that. So once I have got a variance, obviously, I can get the square root of that. You can use the square root function that we have discussed earlier in one of the lecture or you can use the exponent operator to get the square root, but R has inbuilt standard deviation function also.

So that is called sd function, sd(), as I have shown here. So, I will use that with the argument because I want to calculate the standard deviation for systolic blood pressure data, so the argument will be the sys variable. So, that is what I do. So the standard deviation is 5.437524. So, what I have done, I have calculated the mean, median, variance and all these things.

Now, you may be wondering that, is not there a way that I can actually get all these general statistics which are very handy in data analysis? For all these three variables, I have three variables in my data. One is the ages of each of these individual, then their systolic blood pressure and their diastolic blood pressure.

Cannot I somehow get the statistics the descriptive statistics of all these three variables, but just one stroke? Yes, I can get that, but to do that, I have to create a table. So, in Excel, you may have created a table where in this case you will have the first column may be ages, second column may be systolic blood pressure for individual and the third may be the diastolic pressure. So I will create a table in R using this variable ages, dia and sys. So how should I proceed?

(Refer Slide Time: 09:36)

The screenshot shows the RStudio interface. The script editor contains the following code:

```
10 # minimum and maximum values
24 # Median, mean, variance and std. dev
37 # Create a table & get comprehensive statistics ----
38
39 #Table using data.frame() function
40
41 # A data frame has row and columns, just like a table
42
43 bp.data <- data.frame(ages, sys, dia)
44
45 # Get the statistics
```

The console shows the following output:

```
[1] 120
> mean(sys)
[1] 121.3
> var(sys)
[1] 29.56667
> sd(sys)
[1] 5.437524
> bp.data <- data.frame(ages, sys, dia)
>
```

The Environment pane shows the variable `bp.data` with 10 observations of 3 variables. The Plots pane is empty.

The screenshot shows the RStudio interface with the data frame `bp.data` viewed in a table format. The table has 10 rows and 3 columns: `ages`, `sys`, and `dia`.

	ages	sys	dia
1	23	122	83
2	34	120	79
3	35	120	78
4	29	115	72
5	41	130	90
6	42	131	90

The Environment pane shows the variable `bp.data` with 10 observations of 3 variables. The Plots pane is empty.

The console shows the following output:

```
[1] 121.3
> var(sys)
[1] 29.56667
> sd(sys)
[1] 5.437524
> bp.data <- data.frame(ages, sys, dia)
> View(bp.data)
> View(bp.data)
>
```

What I will use I will use a concept called data frame. Let me briefly say, what is data frame. Data frame is essentially something like a table and you must be habituated to creating table and in Excel or any other spreadsheet software you create tables where you may arrange the data each variable in one column and the rows are each sample. So, here also we create something like that a data type called data frame.

And if you think carefully a table is nothing but a matrix, the difference between matrix and a data frame would be, in matrix, if I create a matrix, two-dimensional matrix with suppose ten row three variable, then all the elements in that matrix must be of same type, either they are integer or they are real number or they are character something like that, you cannot mix

them up. Whereas when I can create a table type structure using data frame, each column can have their own data type.

So, I can have one column where only characters name will be there, where I can have another column where I may have numerical values like integers. So, in this case, in this problem, what I want to get I want to create a table or data frame using a data frame function the function itself is called data dot frame. And I will pass this variable, I have three variables, in the environment section you can see I have three variable ages, dia and sys.

```
bp.data ← data.frame(ages, sys, dia)
```

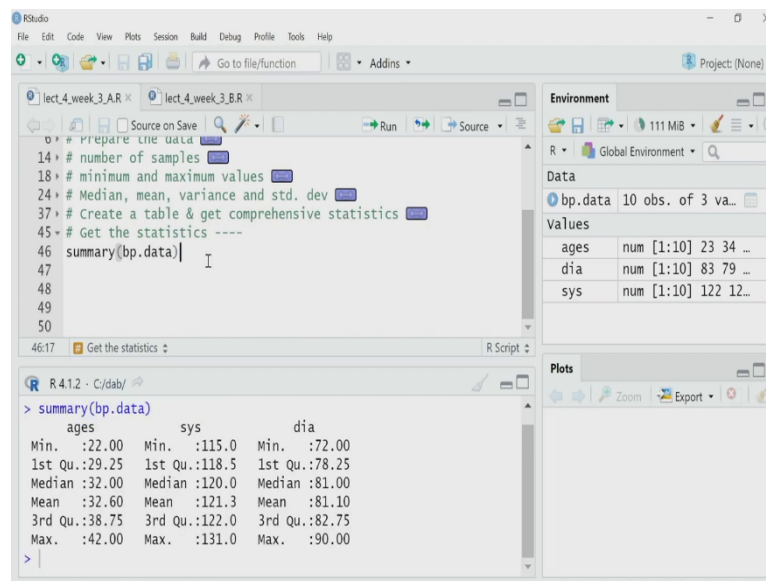
I will pass these three as arguments to this data frame function, data.frame function to create a table with columns corresponding to each of these variables. Let me do it, it will be easy to understand what we are doing. So, what I am using the function that I am using is data.frame as it is written here. And the arguments are the name of the variables, I have three variable ages, sys, and dia. So, those three arguments are comma separated and given inside the round bracket.

And this whole output of this function is assigned to a variable, which I have named as blood pressure data, bp.data. So, let me execute that. Now, look into the environment section. I have a new data type has come it is bp.data, it is a variable and it is saying it has 10 observation and 3 variable. So, let me open this double click it and I can see the data here.

See, I have got what I wanted, I have now a table with the first column having the ages data, age of individual. And then the second column and third column gives me the systolic and diastolic pressure for individual of them. So, I have got the table the way I want it. Now, I want to extract the statistics, the descriptive statistic mean, variance something like that, for this whole thing. So, to do that, let me get back to my script.



(Refer Slide Time: 12:41)



The screenshot shows the RStudio interface. The main editor window contains the following R code:

```
14 # Prepare the data
18 # minimum and maximum values
24 # Median, mean, variance and std. dev
37 # Create a table & get comprehensive statistics
45 # Get the statistics ----
46 summary(bp.data)
47
48
49
50
```

The Environment pane on the right shows the variable `bp.data` with 10 observations and 3 variables: `ages`, `dia`, and `sys`.

The R Console at the bottom shows the output of the `summary(bp.data)` command:

```
> summary(bp.data)
  ages      sys      dia
Min.   :22.00  Min.   :115.0  Min.   :72.00
1st Qu.:29.25  1st Qu.:118.5  1st Qu.:78.25
Median :32.00  Median :120.0  Median :81.00
Mean   :32.60  Mean   :121.3  Mean   :81.10
3rd Qu.:38.75  3rd Qu.:122.0  3rd Qu.:82.75
Max.   :42.00  Max.   :131.0  Max.   :90.00
```

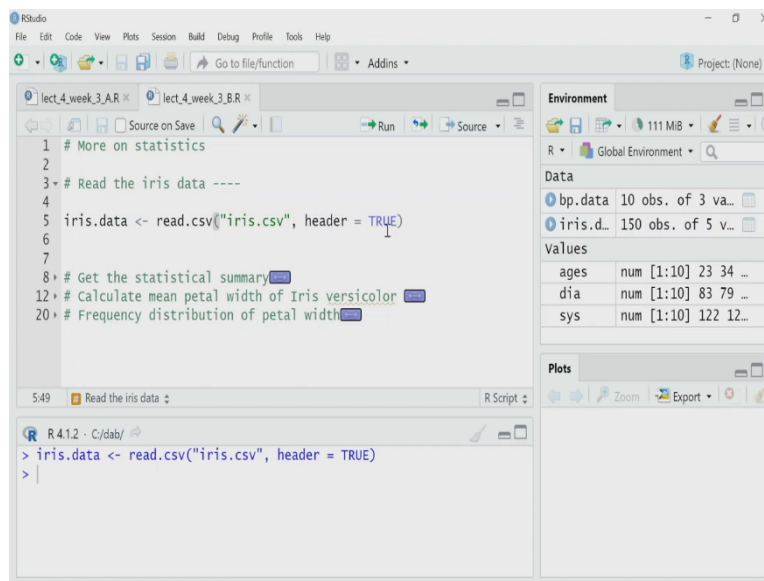
So, I have created the data. Now, what I will do, I will get the statistics on that. So, to get the statistics, I will use with a very handy function, useful function called `summary`. So, if I call this function and if I use this `bp dot data` as the argument then it will give me the summary of this particular table or data frame. So, let me execute it. Here you can see the summary.

`summary(bp.data)`

The summary is like this, for each column, it has written the name of the column, the variable name or the header. So I have `ages`, `sys` and `dia`. And then the rows are certain statistics, which are very commonly useful. Suppose it gives a minimum value and the maximum value for `ages` the minimum value is 22. So, the youngest person in my data set is 22 years old, and the maximum is 42. The oldest person is 42 years old.

It also reports the median and mean for all these three variable, for example, the median and mean for age for these people are same 32. So, this is how you can actually generate descriptive statistics, the basic information basic statistics, which are required to understand the data that I have in hand, before I jump start doing some complicated analysis, there are few other analysis that I can do, let us discuss those.

(Refer Slide Time: 14:11)

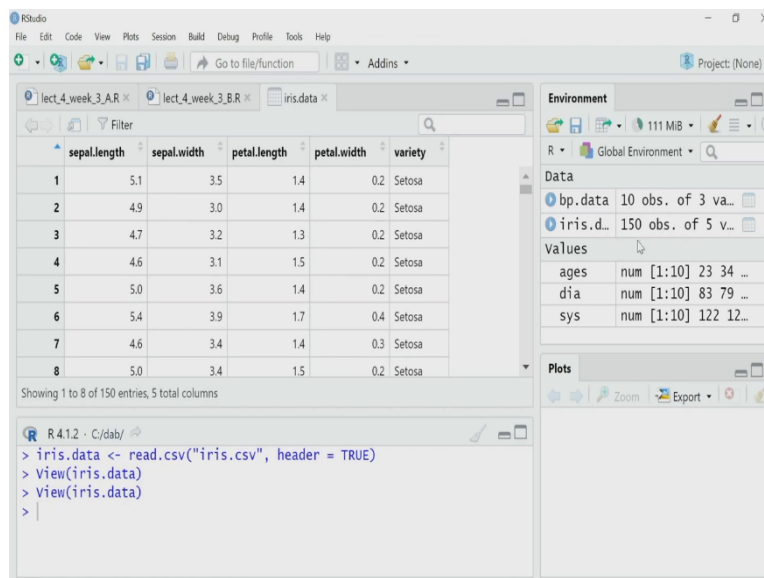


The screenshot shows the RStudio interface with a script editor containing the following R code:

```
1 # More on statistics
2
3 # Read the iris data ----
4
5 iris.data <- read.csv("iris.csv", header = TRUE)
6
7
8 # Get the statistical summary
12 # Calculate mean petal width of Iris versicolor
20 # Frequency distribution of petal width
```

The Environment pane on the right shows the global environment with the following data objects:

Object	Class	Dimensions	Values
bp.data	data.frame	10 obs. of 3 variables	
iris.d...	data.frame	150 obs. of 5 variables	
ages	numeric	[1:10]	23 34 ...
dia	numeric	[1:10]	83 79 ...
sys	numeric	[1:10]	122 12...



The screenshot shows the RStudio interface with the 'iris.data' object selected in the Environment pane. The Data Viewer shows the first 8 rows of the data:

	sepal.length	sepal.width	petal.length	petal.width	variety
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
3	4.7	3.2	1.3	0.2	Setosa
4	4.6	3.1	1.5	0.2	Setosa
5	5.0	3.6	1.4	0.2	Setosa
6	5.4	3.9	1.7	0.4	Setosa
7	4.6	3.4	1.4	0.3	Setosa
8	5.0	3.4	1.5	0.2	Setosa

The console shows the following R commands:

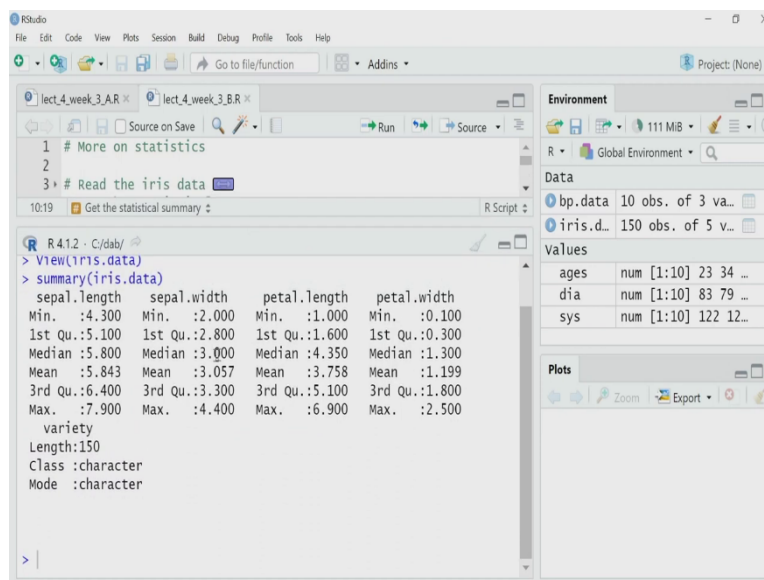
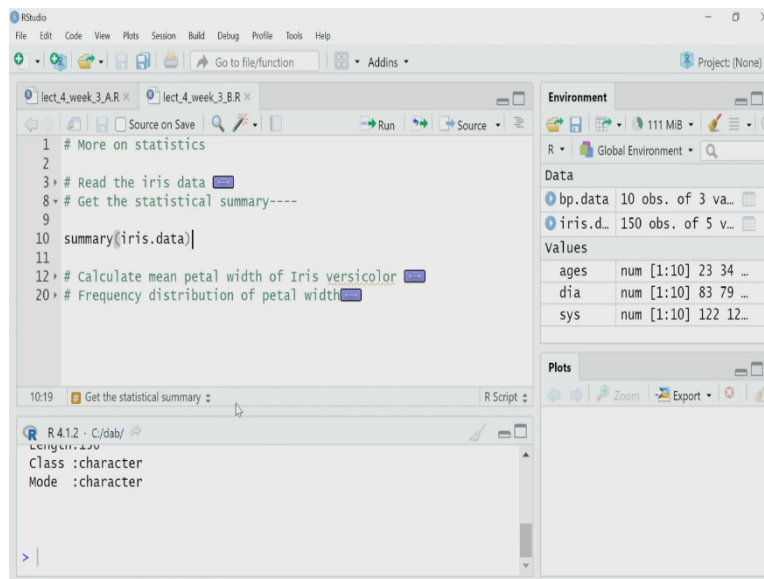
```
> iris.data <- read.csv("iris.csv", header = TRUE)
> View(iris.data)
> View(iris.data)
> |
```

So in this example, the second example what I will do, I will read that iris data, I have the it is in CSV file, we have used that CSV file earlier. So, I will import that and read that CSV data and calculate the summary of that and then also what I will do, I will calculate some basic statistics on these data, sub part of that data. And another interesting thing, I will see the distribution of that data, let us go to that.

So the first thing what I will do, I will read the iris data, iris flower data using the read CSV function. We have done that earlier. So, read CSV, we will take here two arguments I will use. One is the name of the file that is iris dot CSV within the apostrophe and I will specify header is equal to TRUE, and we will assign this whole thing to iris.data variable.

```
iris.data ← read.csv("iris.csv", header = TRUE)
```

(Refer Slide Time: 15:31)

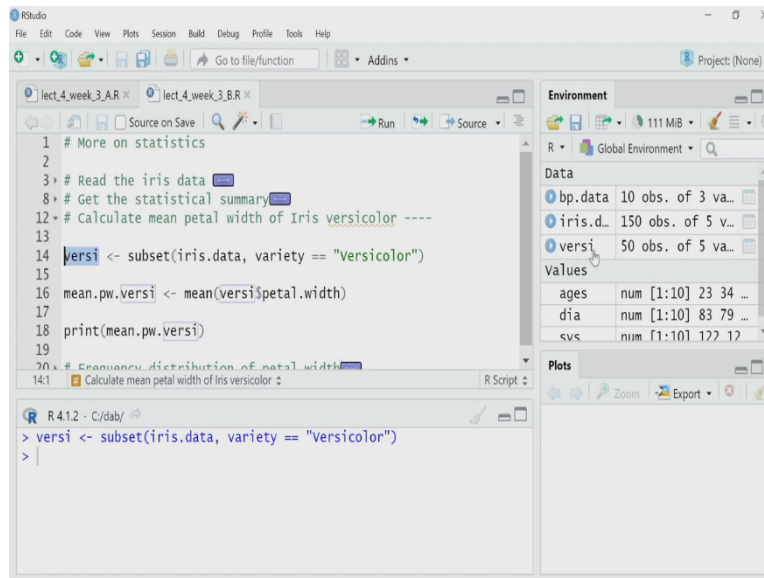


Now, what I want to do, I want to get the statistical summary of that, descriptive statistics of that. So, again, I will call that useful function called `summary`. Now, it will summarize the `iris.data`. So, here comes my summary. So, again each of these variables are in column and it is reporting the minimum, maximum as you can see the mean width sepal width is 2, whereas, the maximum sepal width is 4.4, the median is 3 for sepal width.

Whereas, the median for petal length is 4.35 and the third column is the third variable, sorry not the third, the last variable is variety and it has characters in it. So, you cannot have this

type of statistics for these. Fine. So, I have got the summary of that, but I want to calculate the mean of a particular variable or a subset of those variable, let us do that.

(Refer Slide Time: 16:37)



The screenshot shows the RStudio interface with the following code in the editor:

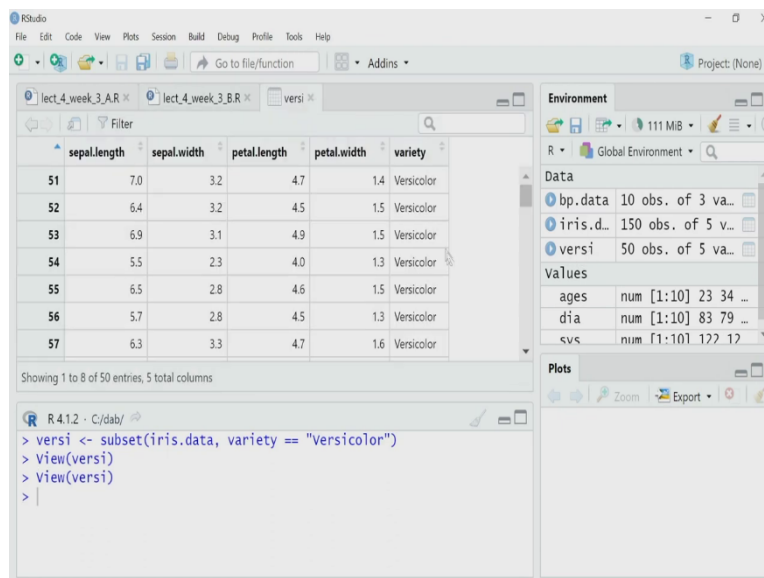
```
1 # More on statistics
2
3 # Read the iris data
8 # Get the statistical summary
12 # Calculate mean petal width of Iris versicolor ----
13
14 vers1 <- subset(iris.data, variety == "versicolor")
15
16 mean.pw.vers1 <- mean(vers1$petal.width)
17
18 print(mean.pw.vers1)
19
20 # Frequency distribution of petal width
14:1 Calculate mean petal width of iris versicolor
```

The Environment pane on the right shows the following objects:

- bp.data: 10 obs. of 3 va...
- iris.d.: 150 obs. of 5 v...
- vers1: 50 obs. of 5 va...

The console shows the execution of the code:

```
R 4.1.2 · C:/dab/
> vers1 <- subset(iris.data, variety == "versicolor")
> |
```



The screenshot shows the RStudio interface with the 'vers1' data frame viewed in a table. The table has 5 columns: sepal.length, sepal.width, petal.length, petal.width, and variety. The data is as follows:

	sepal.length	sepal.width	petal.length	petal.width	variety
51	7.0	3.2	4.7	1.4	Versicolor
52	6.4	3.2	4.5	1.5	Versicolor
53	6.9	3.1	4.9	1.5	Versicolor
54	5.5	2.3	4.0	1.3	Versicolor
55	6.5	2.8	4.6	1.5	Versicolor
56	5.7	2.8	4.5	1.3	Versicolor
57	6.3	3.3	4.7	1.6	Versicolor

The console shows the execution of the code:

```
R 4.1.2 · C:/dab/
> vers1 <- subset(iris.data, variety == "versicolor")
> view(vers1)
> View(vers1)
> |
```

```

1 # More on statistics
2
3 # Read the iris data
4 # Get the statistical summary
5 # Calculate mean petal width of Iris versicolor ----
6
7
8
9
10
11
12 # Calculate mean petal width of Iris versicolor ----
13
14 versicolor <- subset(iris.data, variety == "versicolor")
15
16 mean.pw.versicolor <- mean(versicolor$petal.width)
17
18 print(mean.pw.versicolor)
19
20 # Frequency distribution of petal width
21
22 Calculate mean petal width of Iris versicolor

```

Environment

Object	Class	Attributes
bp.data	data.frame	10 obs. of 3 va...
iris.d.	data.frame	150 obs. of 5 v...
versicolor	data.frame	50 obs. of 5 va...

Values

ages	num	[1:10] 23 34 ...
dia	num	[1:10] 83 79 ...
mean.pw.versicolor	num	1.326

```

> View(versicolor)
> View(versicolor)
> View(versicolor)
> mean.pw.versicolor <- mean(versicolor$petal.width)
> mean.pw.versicolor
[1] 1.326
> |

```

`versicolor <- subset(iris.data, variety == "versicolor")`

So, what I will do now, I will calculate the mean petal width of only one type of flower that is iris versicolor. So, they are if you remember in this data set, iris data set, there are multiple iris flowers data are there in the one single file. So, that is defined by the variable variety. And I want to know the mean petal width of iris versicolor, only one type of, one variety of the flower.

So, in this, to do this, what I will do, I will extract only the data for the versicolor variety and then I will calculate the statistics. So, to do the subset of the data, I will use the subset function that we have learned earlier. So, subset will give me a subset of the whole data and it will take the arguments will for these will be iris.data is the name of the data.

So, that is one argument and the second argument which is a logical argument, what I am telling it here, I am saying the variety variable must be equal to, two equal to sign is there, is a logical equal. So, this equal to what versicolor. And versicolor is a character, so that is why it has been written within the apostrophe. So, if I run this code, I create a new data set versicolor because I have assigned these to this versicolor variable. So, let me open that.

So, you can see now I have 50 observations and all of these are a variety versicolor. Because remember, I have used the variety equal to versicolor as a logic to extract the subset. So, I have extracted that. Now, I want to calculate the mean of the petal width of this versicolor flowers.

So, what I will do, if I check these versicolor data, this petal width is one variable, so I can extract that one using the dollar sign(\$), dollar operator.

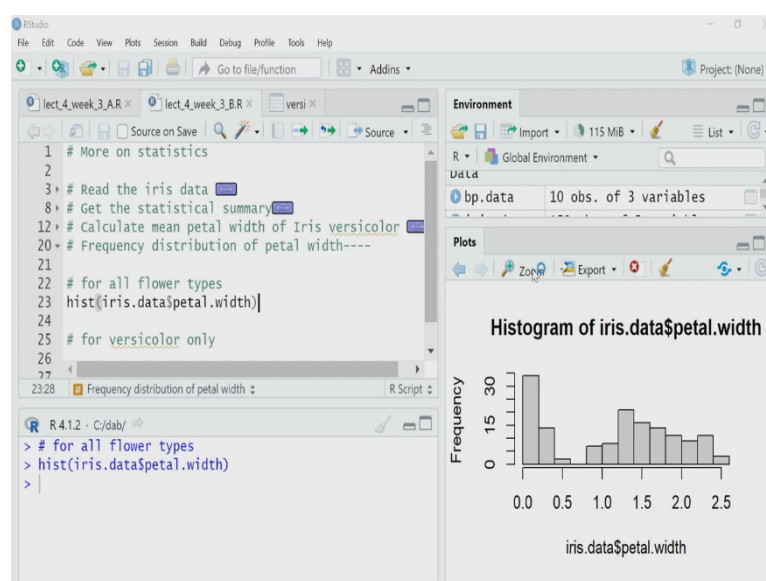
```
mean.pw.versi ← mean(versi$petal.width)
```

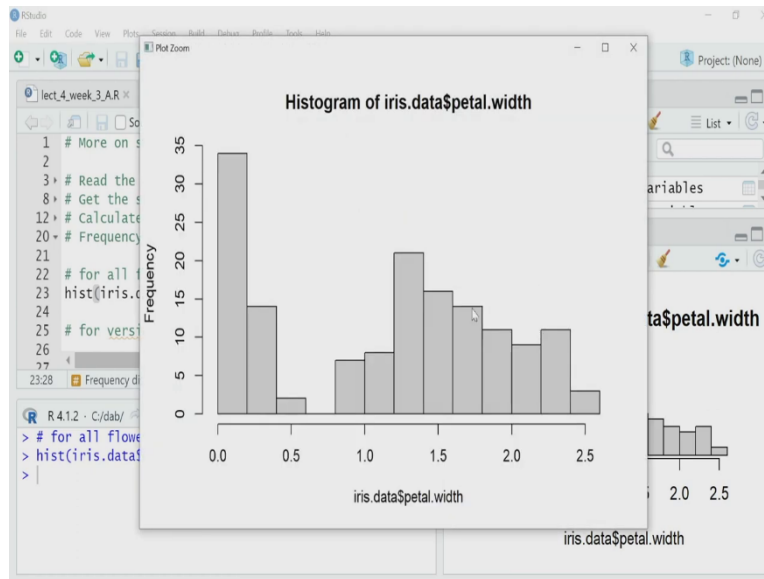
So, that is what I am doing here. I am calling a mean function to calculate the mean. Now, the argument is `versi$petal.width`. Now, `petal.width` is a variable of `versi` data and I am asking using this dollar sign that I want only that column that variable that object. So, I want one that part only and I want to calculate the mean of that.

So, what I am asking is that you take this petal width column only, petal width variable only and calculate the mean of that. So, let me run that. Yes, it has calculated and that mean I have assigned as `mean.pw.versi`. So, let me check out what will be the value of that, `mean.pw.versi`, the value is 1.326. The last thing that I want to discuss in this video is that apart from these numerical value, these basic statistics in terms of numbers, many a times I want to visually see the distribution.

Because remember some time it may happen that in your data there are lots of outlier. Some sample has very high value or very low value, but your mean, median may not reflect that. So, it is always good before you jump into any analysis, why do not you plot a frequency histogram. So, what I will show to you now is that how to create a frequency histogram for my data.

(Refer Slide Time: 20:27)





`hist(iris.data$petal.width)`

So, what I will do I will calculate the frequency, draw the frequency histogram for the petal width, I could do that for any other variable in my this data set, but I will do it for petal width first. So, to do that, I will use the function called hist, hist function create frequency histogram by default. It has lots of argument by which you can actually create the histogram, but I will use the basic one that I will give the name of the data variable to it.

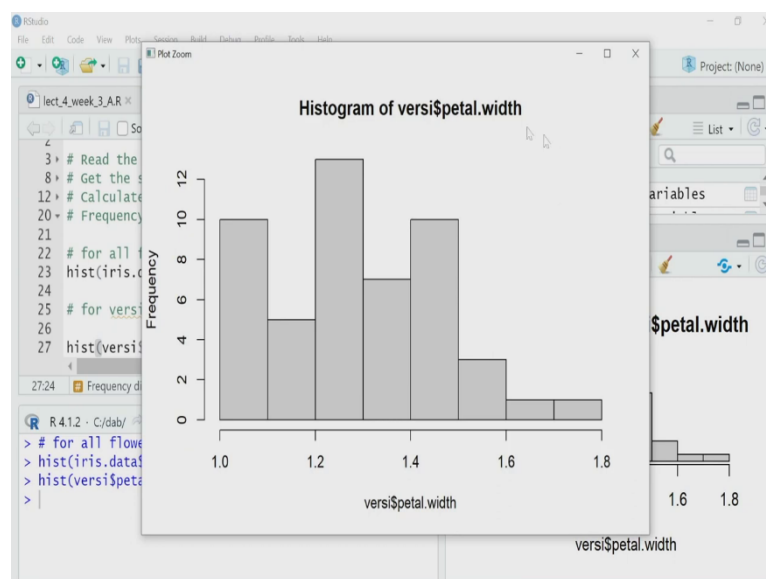
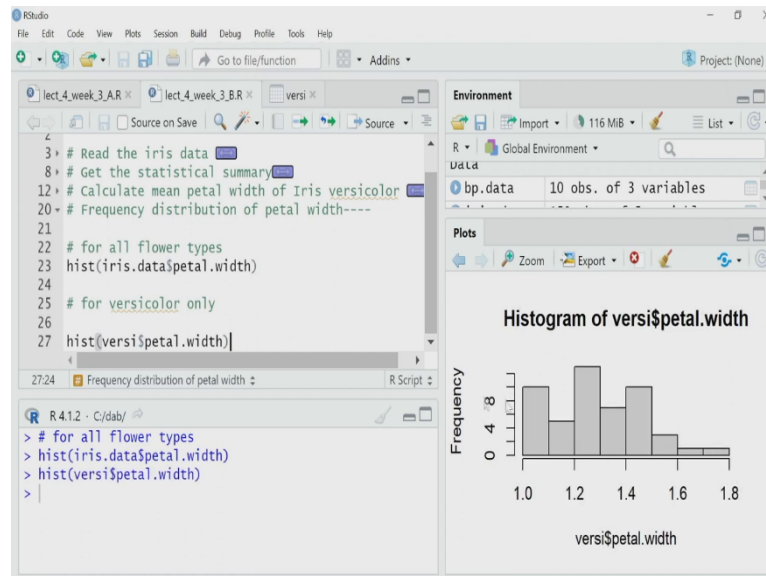
I will just give the data as the single argument to this hist function and it will draw the histogram by default options. So, what will be my data my original data set is iris.data. And I want to get only the petal width variable. So, I am again using the dollar sign here. So, iris.data then I am putting this dollar operator and then I am asking the for the variable petal.width. So, it will give me, it will take only the petal.width variable data and it will draw the histogram for that.

So, here I get the histogram let me zoom it, so, that it will be clear for you using these zoom option given here, I could zoom it. You can see by default options it has drawn the histogram. So, the horizontal axis is the sepal width, I have not given any specific names. So, it is just taking the name of the variable directly. And on the vertical axis I have the frequency rather than the count actually.

Now, you can easily see I have a mixed population. Obviously, I have different types of, different varieties of flower, iris flower in that and they are petal width of each all of them are not similar. So, I have distinct behavior here there are some flower which may belong to this lower part of the histogram, whereas, the other flowers may belong to these extended one the

larger one. So, now, what I will do, I will try to calculate the histogram I will try to draw the histogram for only the versicolor variety suppose.

(Refer Slide Time: 22:34)



hist(versi\$petal.wdith)

So, what I will do, let me close this, I want to draw it for the versicolor. So, already I have extracted one variable, which is called versi earlier, so that versi has my versicolor data and from that I want to extract only the petal width variable. So, again I will use this operator of dollar sign and I will draw the histogram for that variable in the versicolor. So, I execute it using the same hist function with default options.



Now, here again let me zoom it. I have got the histogram for the petal width of only versicolor type, versicolor variety of iris flower and it has largely homogeneous one. You can see it is bit heavy on the left hand side and it is a bit narrow on the higher values. This brings us to the end of this lecture. In just a brief in this lecture we have learned how to calculate the basic statistic which you call descriptive statistics that give us a overall feel of the data using R. Thank you for joining with me today. See you in the next one.