Essentials of Data Science with R Software - 2
Sampling Theory and Linear Regression Analysis

Prof. Shalabh
Department of Mathematics and Statistics
Indian Institute of Technology Kanpur

Sampling Theory with R software
Lecture - 31
Stratified Random Sampling
Drawing of Sample Using sampling and strata Packages in R

Hello, friends welcome to the course Essentials of Data Science with R Software 2, where we are trying to understand the topics of Sampling Theory and Linear Regression Analysis. In this module we are going to continue with Sampling Theory with R Software and we are going to continue with the chapter Stratified Random Sampling with R software. So, you can recall that up to now we have developed the theory part of stratified random sampling.

Now the next question is how to implement those things on the R software? So, as you have seen in the case of simple random sampling also, when the role of software like R comes into picture. Then R is helping us in drawing the suitable sample, and after that we have to write a small function, to find out what we aim to find.

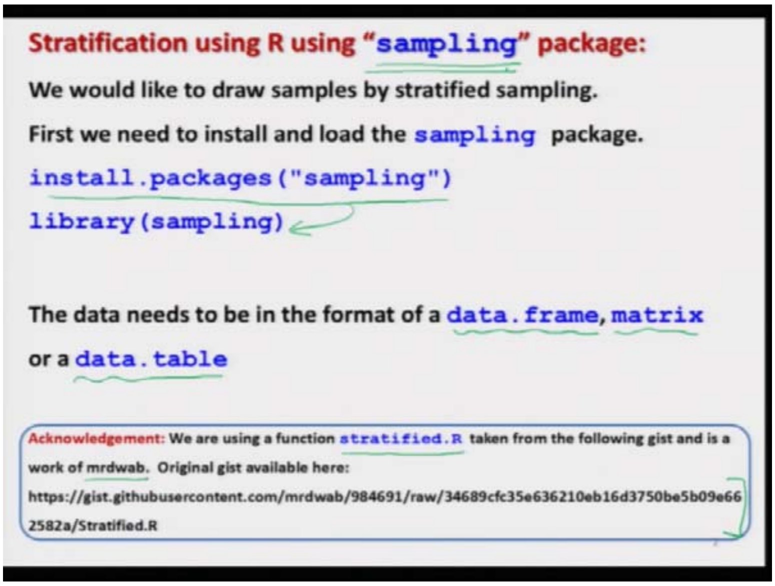
Similar is the story in the case of stratified sampling also. At least to the best of my knowledge I could find three possible packages, which are helping in drawing the sample from a stratified random sampling setup and those packages are sampling, strata and survey. And means all those packages are more or less similar.

The only thing is this they have got different types of commands, different types of options. So, as far as this sampling and strata are concerned they basically help us in drawing the sample only. Once we draw the sample, then we can carry forward our computations further. But this packages survey has one more advantage that it gives us the standard error as well as the estimate of population mean.

So, I will try to give you the details of the, of these three packages and among these three packages. I will be giving you more details on the package of package sampling. And once you understand the sampling package, how to execute it and how to draw the samples, I will try to give you a quick overview of the packages say strata and survey, , right.

So, that is our plan and this is how we try to proceed. So, first I will try to show you the slides and then, I will come to the R software and I will try to execute those commands, which I will teach you ok. So,, right.

(Refer Slide Time: 03:10)



Stratification using R using “sampling” package:

We would like to draw samples by stratified sampling.

First we need to install and load the sampling package.

```
install.packages("sampling")
library(sampling)
```

The data needs to be in the format of a data.frame, matrix or a data.table

Acknowledgement: We are using a function stratified.R taken from the following gist and is a work of mrdwab. Original gist available here: https://gist.githubusercontent.com/mrdwab/984691/raw/34689cfc35e636210eb16d3750be5b09e662582a/Stratified.R

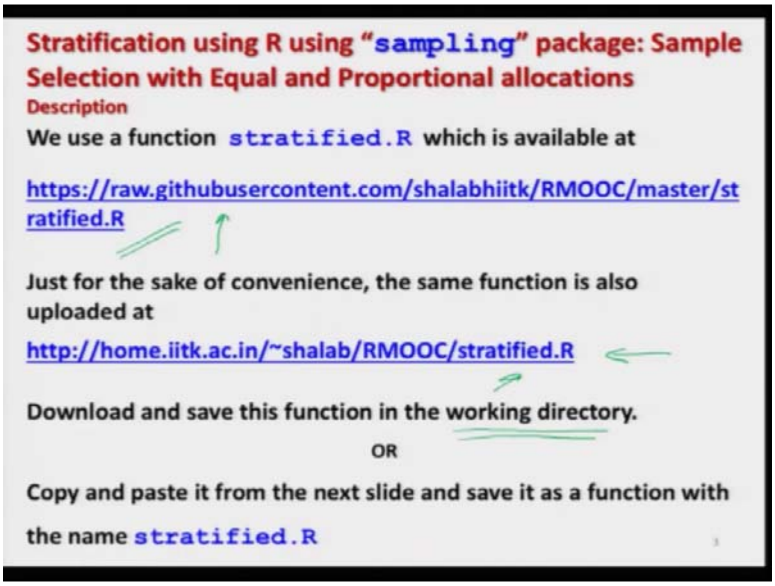
So, in order to use the package sampling, we first need to install it and load it on our computer. So, in order to install the packages we have to use the command, install.packages and within the double quotes inside the parenthesis we have to write sampling. And once you install it, then you can load it by the command library.

One of the conditions and restrictions in this package is that the data has to be in the format of data frame or matrix or a tabular that is data.table format. Now, from this package we have to draw the samples and in order to draw the samples, I am going to use here a function, which is named as a stratified.R.

This is a script, this is a function, which has been written and means I would say make it clear, that I have not written this package, but this package is available in the github. Github is a depository, that is a website and actually this is a work of someone, whose login is mrdwab.

And this script is available at this address and I would like to express my acknowledgement for this person, who has developed this program to draw the sample. And you will see that, this is a pretty simple program and it is very easy to handle.

(Refer Slide Time: 05:00)



Stratification using R using "`sampling`" package: Sample Selection with Equal and Proportional allocations

Description

We use a function `stratified.R` which is available at

https://raw.githubusercontent.com/shalabhiitk/RMOOC/master/stratified.R

Just for the sake of convenience, the same function is also uploaded at

http://home.iitk.ac.in/~shalab/RMOOC/stratified.R

Download and save this function in the working directory.

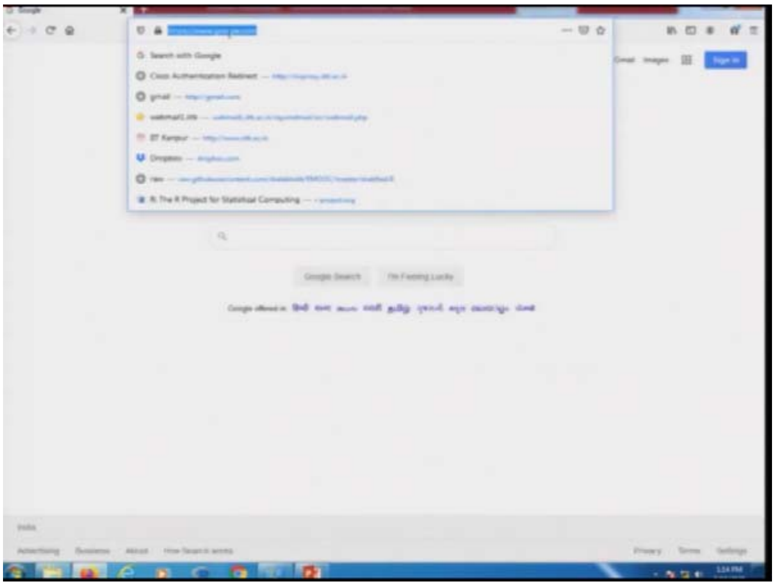
OR

Copy and paste it from the next slide and save it as a function with the name `stratified.R`

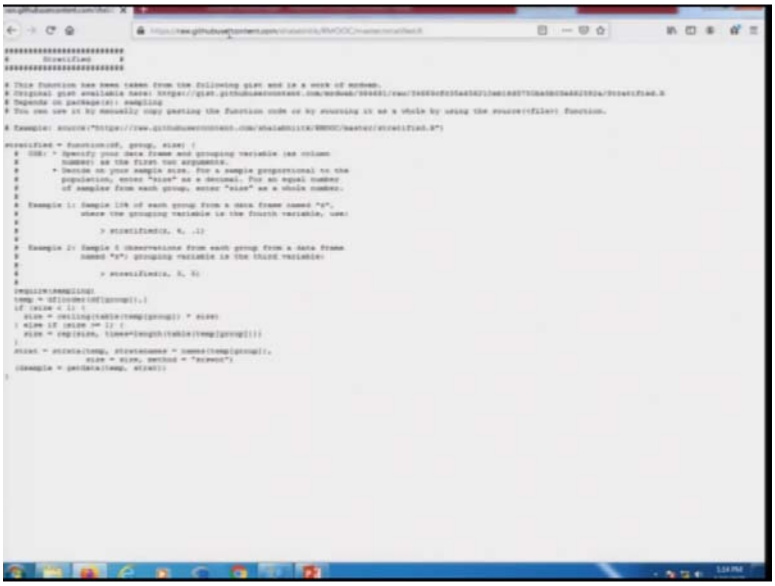
So, if you follow my slides to draw the stratified random sample, first you need to have this function stratified.R on your computer. So, what I have done that here in my slide, I am giving you this address and this program stratified.R is there. And you simply have to copy this address from here, and paste it any web browser. And this program you can download or you can also copy from there and paste it in the R console.

I also have uploaded this program on my personal website, just for the sake of convenience. So, you can also copy and paste this program from here also, right and you have to download this program. And you have to save it in the working directory, in which you in which all your programs are stored or all the files are stored, when you try to work on the R software.

(Refer Slide Time: 06:13)



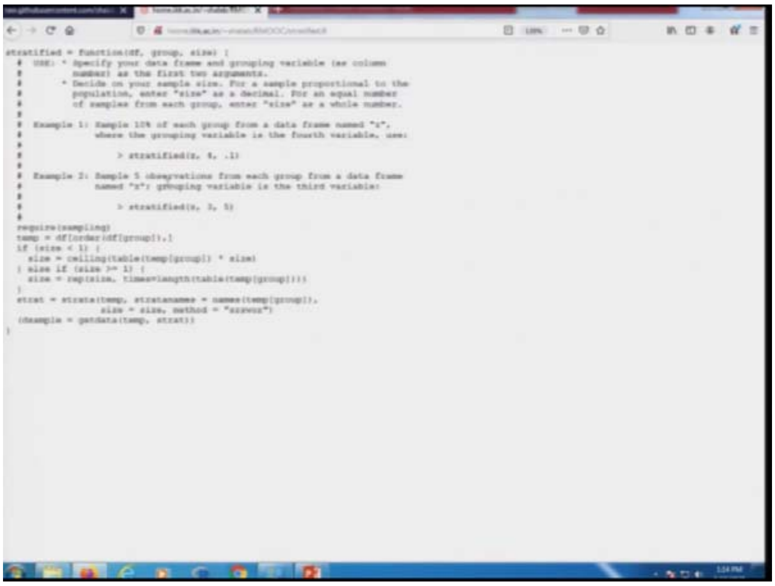
(Refer Slide Time: 06:16)



```
###########################
#       Stratified        #
###########################

# This function has been taken from the following gist and is a work of mrdwab.
# Original gist available here: https://gist.githubusercontent.com/mrdwab/984481/raw/[illegible]/Stratified.R
# Depends on package(s): sampling
# You can use it by manually copy pasting the function code or by sourcing it as a whole by using the source(<file>) function.

# Example: source("https://raw.githubusercontent.com/shalabhiitk/RMOOC/master/stratified.R")

stratified = function(df, group, size) {
  #  USE: * Specify your data frame and grouping variable (as column
  #         number) as the first two arguments.
  #       * Decide on your sample size. For a sample proportional to the
  #         population, enter "size" as a decimal. For an equal number
  #         of samples from each group, enter "size" as a whole number.
  #
  #  Example 1: Sample 10% of each group from a data frame named "z",
  #             where the grouping variable is the fourth variable, use:
  #
  #                 > stratified(z, 4, .1)
  #
  #  Example 2: Sample 5 observations from each group from a data frame
  #             named "z"; grouping variable is the third variable:
  #
  #                 > stratified(z, 3, 5)
  #
  require(sampling)
  temp = df[order(df[group]),]
  if (size < 1) {
    size = ceiling(table(temp[group]) * size)
  } else if (size >= 1) {
    size = rep(size, times=length(table(temp[group])))
  }
  strat = strata(temp, stratanames = names(temp[group]),
                 size = size, method = "srswor")
  (dsample = getdata(temp, strat))
}
```

So, just to show you here, how it is going to work means I can copy this address over here and I come to here web browser. And I try to paste it here you can see here this is the program which is here.

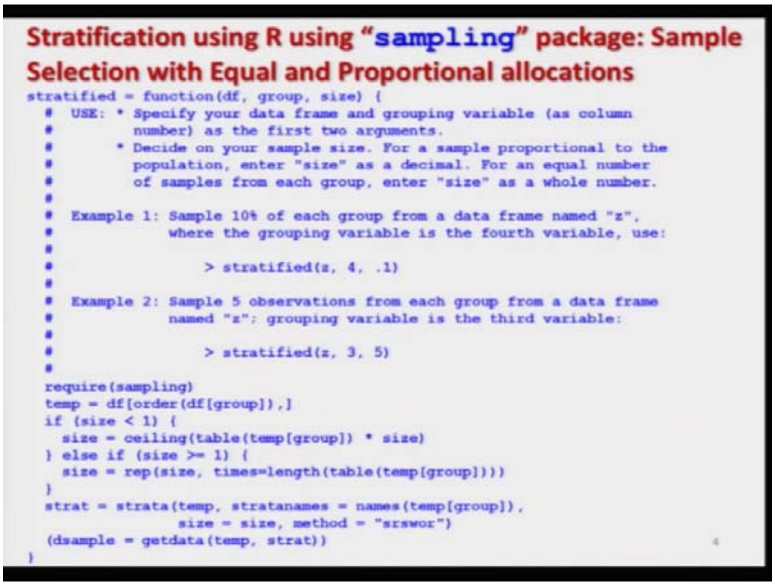
(Refer Slide Time: 06:34)



```
stratified = function(df, group, size) {
  #  USE: * Specify your data frame and grouping variable (as column
  #         number) as the first two arguments.
  #       * Decide on your sample size. For a sample proportional to the
  #         population, enter "size" as a decimal. For an equal number
  #         of samples from each group, enter "size" as a whole number.
  #
  #  Example 1: Sample 10% of each group from a data frame named "z",
  #             where the grouping variable is the fourth variable, use:
  #
  #                 > stratified(z, 4, .1)
  #
  #  Example 2: Sample 5 observations from each group from a data frame
  #             named "z"; grouping variable is the third variable:
  #
  #                 > stratified(z, 3, 5)
  #
  require(sampling)
  temp = df[order(df[group]),]
  if (size < 1) {
    size = ceiling(table(temp[group]) * size)
  } else if (size >= 1) {
    size = rep(size, times=length(table(temp[group])))
  }
  strat = strata(temp, stratanames = names(temp[group]),
                 size = size, method = "srswor")
  (dsample = getdata(temp, strat))
}
```

So, you can simply copy it or if you want to see from this site also. I can copy means another address just to show you, you can see here this is here,, right. So, you can simply just copy from here and just simply paste it into your R console that is what I want.

(Refer Slide Time: 06:49)



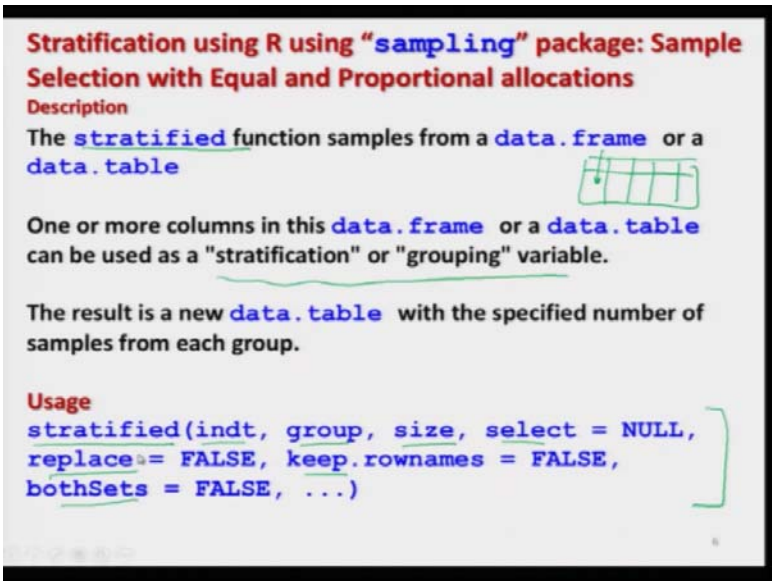
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

```
stratified = function(df, group, size) {
  #  USE: * Specify your data frame and grouping variable (as column
  #         number) as the first two arguments.
  #       * Decide on your sample size. For a sample proportional to the
  #         population, enter "size" as a decimal. For an equal number
  #         of samples from each group, enter "size" as a whole number.
  #
  #  Example 1: Sample 10% of each group from a data frame named "z",
  #             where the grouping variable is the fourth variable, use:
  #
  #                 > stratified(z, 4, .1)
  #
  #  Example 2: Sample 5 observations from each group from a data frame
  #             named "z"; grouping variable is the third variable:
  #
  #                 > stratified(z, 3, 5)
  #
  require(sampling)
  temp = df[order(df[group]),]
  if (size < 1) {
    size = ceiling(table(temp[group]) * size)
  } else if (size >= 1) {
    size = rep(size, times=length(table(temp[group])))
  }
  strat = strata(temp, stratanames = names(temp[group]),
                 size = size, method = "srswor")
  (dsample = getdata(temp, strat))
}
```

4

And just for your better understanding, I have copied this program here also. So, you can also copy it from my slide and you can proceed forward. And here I am giving you a screenshot, that how it is uploaded on my website.

(Refer Slide Time: 07:06)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Description

The stratified function samples from a data.frame or a data.table

One or more columns in this data.frame or a data.table can be used as a "stratification" or "grouping" variable.

The result is a new data.table with the specified number of samples from each group.

Usage

```
stratified(indt, group, size, select = NULL,
replace = FALSE, keep.rownames = FALSE,
bothSets = FALSE, ...)
```

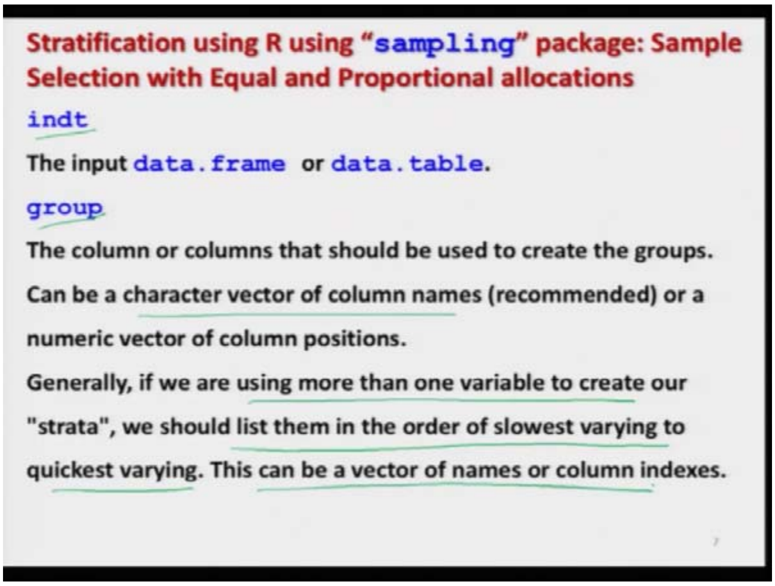
.

And once you have this program on your computer and you have uploaded it on your R console, then we have to use stratified function to sample from the given data frame or a given data table, right. Now, the question is this, how are you going to sample? When because the data frame will have different columns. For example, a data set may look like that there are different variables, which are arranged in different columns.

So, first you need to specify that, which is your grouping variable or which is the variable which you are going to use for stratification. So, one of the one or more columns in this data frame is to be used as a stratification or grouping variable. And then the command to draw the sample is given here, you have to write down stratified.

And after that there are several options, like as indt, group, size, select, replace, keep, both sets etc. So, I will try to give you a brief idea that, what is the role of these options, right.

(Refer Slide Time: 08:19)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

indt

The input data.frame or data.table.

group

The column or columns that should be used to create the groups.

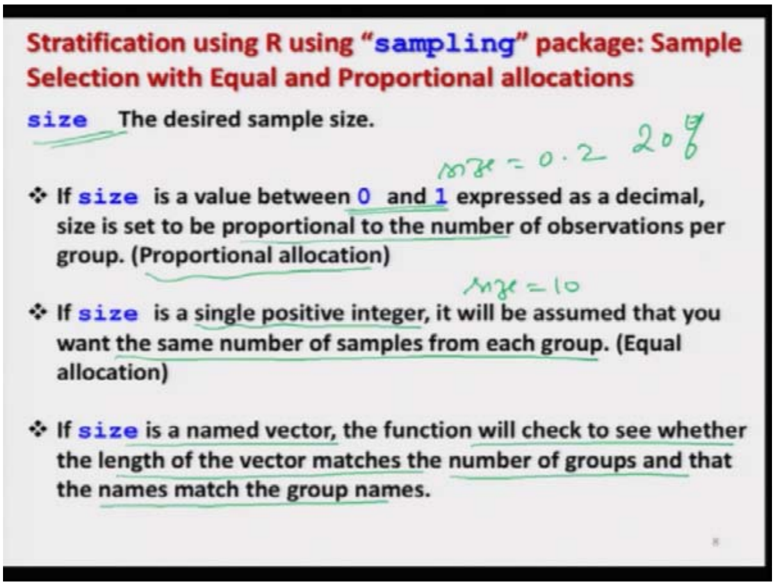
Can be a character vector of column names (recommended) or a numeric vector of column positions.

Generally, if we are using more than one variable to create our "strata", we should list them in the order of slowest varying to quickest varying. This can be a vector of names or column indexes.

This value here indt, this is the input data. Here you have to specify that, what is your data frame or data table, from where you want to draw the samples. Then there is another option here group, this group is actually column which has to be used to create the groups.

So, this can be a character vector of column names or a numeric vector of some column positions. Actually if we are using more than one variable to create our strata, then we should list them in the order of slowest varying to the quickest varying. So, this can be a vector of names or column indexes, right.

(Refer Slide Time: 09:07)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

size The desired sample size.

size = 0.2 20%

- If **size** is a value between 0 and 1 expressed as a decimal, size is set to be proportional to the number of observations per group. (Proportional allocation)

size = 10

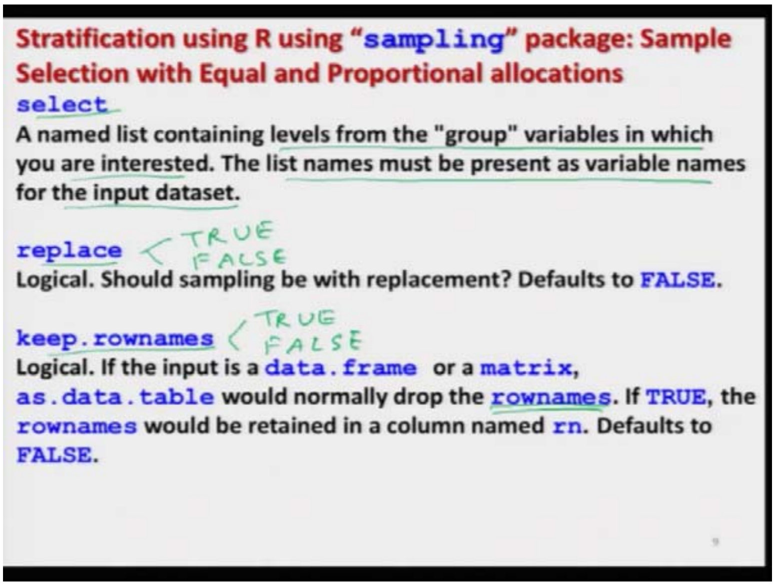
- If **size** is a single positive integer, it will be assumed that you want the same number of samples from each group. (Equal allocation)
- If **size** is a named vector, the function will check to see whether the length of the vector matches the number of groups and that the names match the group names.

Then there is an option here size. So, this is the desired disk sample size. So, here in this package there are two option that, we can draw the samples by equal allocation or by proportional allocation. So, if I try to give the value of size between 0 and 1 and say express as a decimal, then the size is set up to the proportional allocation. For example, if I say size is equal to 0.2; say size is equal to 0.2.

So, that is indicating that we have to draw 20 percent of the sample from every strata. So, this is essentially your proportional allocation. And if you try to give this size as a single positive integers, for example, I can say here size is equal to here, 10 then it will indicate the equal allocation and this will indicate that we want to have the same number of sample, from each of the group that each of the stratum.

And similarly. if a size is a named vector, then the function will actually check to see whether the length of the vector matches the number of groups and the names may match the group names, right ok.

(Refer Slide Time: 10:35)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

select

A named list containing levels from the "group" variables in which you are interested. The list names must be present as variable names for the input dataset.

replace < TRUE FALSE

Logical. Should sampling be with replacement? Defaults to FALSE.

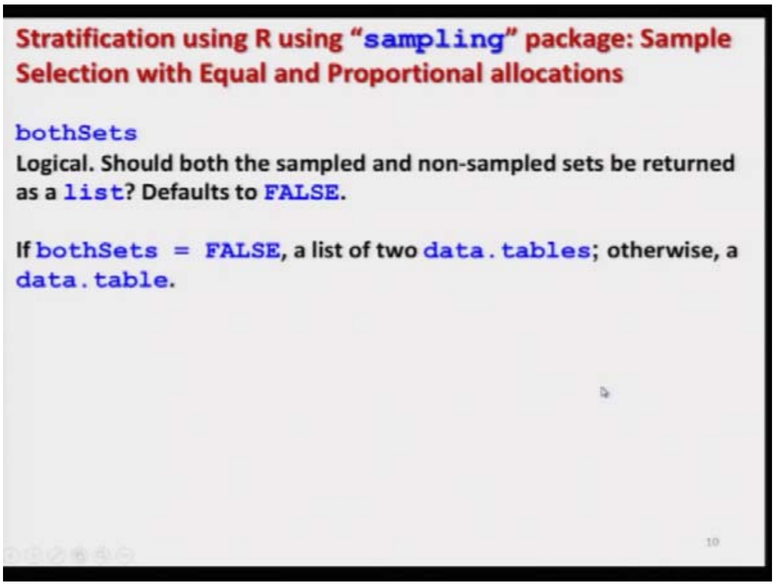
keep.rownames < TRUE FALSE

Logical. If the input is a data.frame or a matrix, as.data.table would normally drop the rownames. If TRUE, the rownames would be retained in a column named rn. Defaults to FALSE.

Now, after this there is another option here select. So, this select is a list which contains the level from the group variable, in which we are interested. And this list name must be presented as a variable name for the input data set. Then we have an option here replace. So, it will indicate that whether we want to draw the sample by srswor or srswr, that is without replacement or with replacement. And this is a logical option; that means, it will take two possible value TRUE or FALSE.

And similarly there is another option here keep, row name this is also a logical variables. So, this will take two possible values TRUE or FALSE. And usually what happened that the if the input is coming from a data frame or a matrix or a data table? Then usually the row names are dropped, but if you want to keep the row names, then we have to use this option.

(Refer Slide Time: 11:54)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

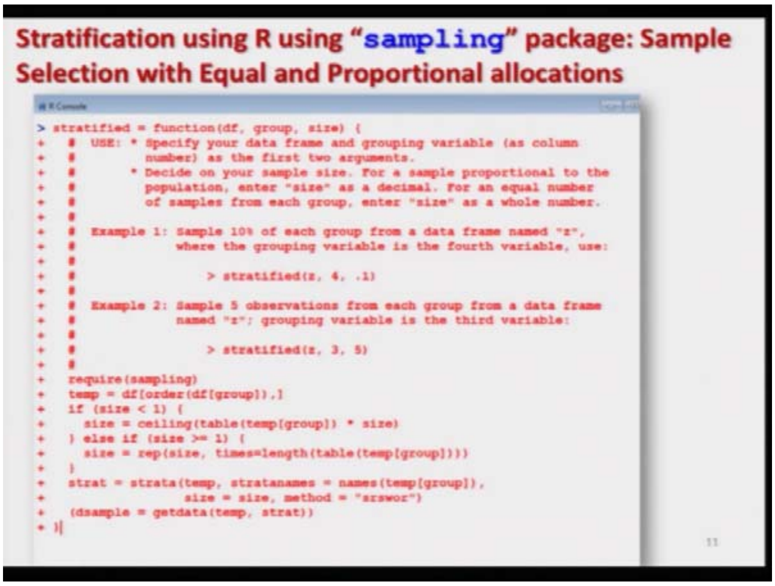
bothSets
Logical. Should both the sampled and non-sampled sets be returned as a list? Defaults to FALSE.

If bothSets = FALSE, a list of two data.tables; otherwise, a data.table.

10

Rather I will take here a simple example, to show you that the things are pretty simple and how these things can be used. Similarly, if there is other option both set, but I will request you that you please go to the help menu and try to see all these options before you move forward.

(Refer Slide Time: 12:13)



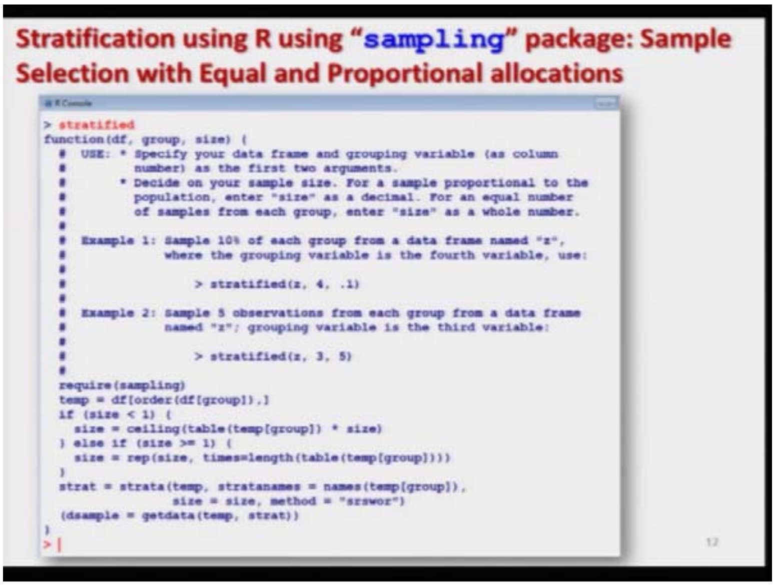
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

```
> stratified = function(df, group, size) {
+   #  USE: * Specify your data frame and grouping variable (as column
+   #         number) as the first two arguments.
+   #       * Decide on your sample size. For a sample proportional to the
+   #         population, enter "size" as a decimal. For an equal number
+   #         of samples from each group, enter "size" as a whole number.
+   #
+   #  Example 1: Sample 10% of each group from a data frame named "z",
+   #             where the grouping variable is the fourth variable, use:
+   #
+   #                 > stratified(z, 4, .1)
+   #
+   #  Example 2: Sample 5 observations from each group from a data frame
+   #             named "z"; grouping variable is the third variable:
+   #
+   #                 > stratified(z, 3, 5)
+   #
+   require(sampling)
+   temp = df[order(df[group]),]
+   if (size < 1) {
+     size = ceiling(table(temp[group]) * size)
+   } else if (size >= 1) {
+     size = rep(size, times=length(table(temp[group])))
+   }
+   strat = strata(temp, stratanames = names(temp[group]),
+                  size = size, method = "srswor")
+   (dsample = getdata(temp, strat))
+ }
```

11

So, here you can see here, I have copied and pasted this program on my R console. I will try to show you on the when I try to execute it, but you can see here this is how it will look like, you simply have to just copy the entire program and just paste it on your R console, right ok.

(Refer Slide Time: 12:35)



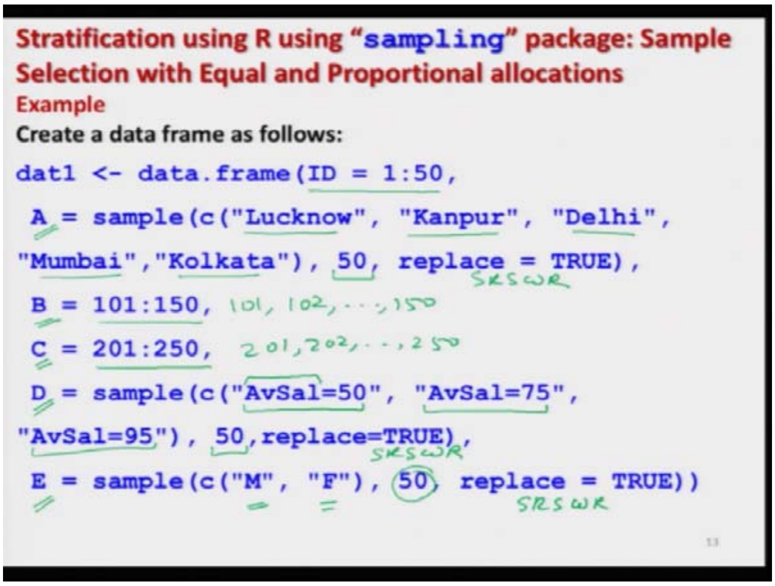
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

```
> stratified
function(df, group, size) {
  #  USE: * Specify your data frame and grouping variable (as column
  #         number) as the first two arguments.
  #       * Decide on your sample size. For a sample proportional to the
  #         population, enter "size" as a decimal. For an equal number
  #         of samples from each group, enter "size" as a whole number.
  #
  #  Example 1: Sample 10% of each group from a data frame named "z",
  #             where the grouping variable is the fourth variable, use:
  #
  #                 > stratified(z, 4, .1)
  #
  #  Example 2: Sample 5 observations from each group from a data frame
  #             named "z"; grouping variable is the third variable:
  #
  #                 > stratified(z, 3, 5)
  #
  require(sampling)
  temp = df[order(df[group]),]
  if (size < 1) {
    size = ceiling(table(temp[group]) * size)
  } else if (size >= 1) {
    size = rep(size, times=length(table(temp[group])))
  }
  strat = strata(temp, stratanames = names(temp[group]),
                 size = size, method = "srswor")
  (dsample = getdata(temp, strat))
}
> |
```

12

And this is the screenshot. So, after you have pasted it correctly ok.

(Refer Slide Time: 12:42)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example

Create a data frame as follows:

```
dat1 <- data.frame(ID = 1:50,
 A = sample(c("Lucknow", "Kanpur", "Delhi",
"Mumbai","Kolkata"), 50, replace = TRUE),
 B = 101:150,
 C = 201:250,
 D = sample(c("AvSal=50", "AvSal=75",
"AvSal=95"), 50,replace=TRUE),
 E = sample(c("M", "F"), 50, replace = TRUE))
```

13

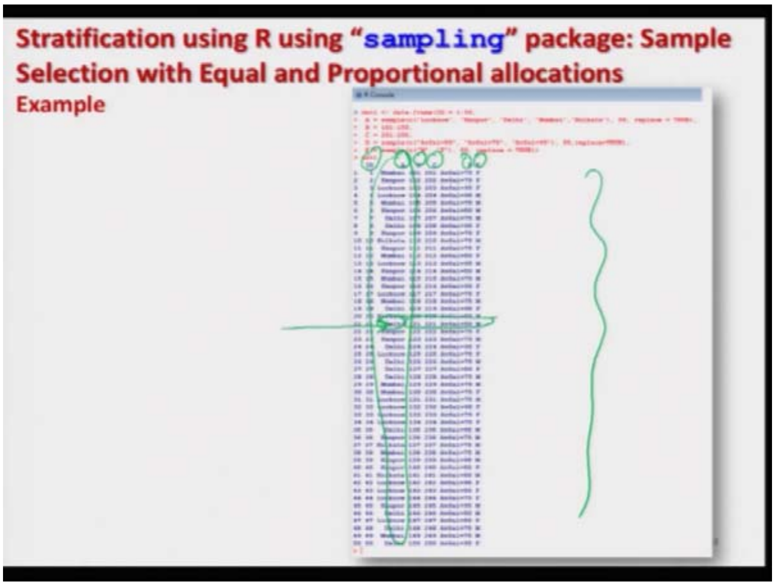
Now, I do. So, now in order to explain you, I have artificially created a data frame of 6 columns. So, this data frame is created in the following way, that the first column is a variable capital ID, which has number from 1 to 50. Then in the second column, I have taken a variable here A in which I have chosen 5's names of the cities Lucknow, Kanpur, Delhi, Mumbai and Kolkata.

And from these 5 cities, we try to draw a sample of size 50 by SRSWR so; that means, we have 50 observations on A. Similarly, I try to choose the second variable B, thus in the column B there are 50 values starting from 101, 102 to up to 150. And similarly in the next column, we have a variable here C, which is taking the value from 201, 202 to 250.

And then the next variable is here D, D consists of 3 values average salary is equal to 50, average salary is equal to 75 and average salary is equal to 95. This average salary is indicated by AvSal and from these 3 values; we try to draw a sample of size 50 by SRSWR, right. And then similarly I try to take the last column E of 2 values, say male and female which are indicated by M and F. And I try to take a sample of 50 observations by SRSWR.

So, you can see here means, I have taken these many variables and on every each of the variable, I am trying to take 50 observation. Because in the data frame, the condition is that every column should have the equal number of observation.

(Refer Slide Time: 15:08)



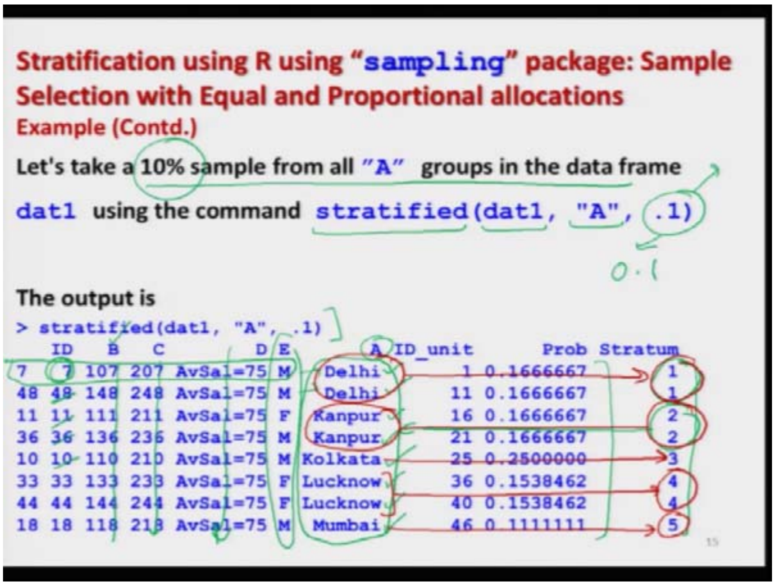
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example

And when I try to create this data set over here, you can see this will look like this. This is a screenshot, which is here ID, then here variable A, variable B, variable C, variable D and variable E, right. Definitely, I am not executing this data set at this moment, because you can see here I am using the command sample here, sample here, sample here, sample here.

So, when I try to draw the sample by simple random sampling with replacement, my data will be different than the sample which I have represented here. So, my data frame will also be different than this data frame. But now I am taking this data frame as fixed quantity.

(Refer Slide Time: 15:51)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

Let's take a 10% sample from all "A" groups in the data frame dat1 using the command stratified(dat1, "A", .1)

The output is

```
> stratified(dat1, "A", .1)
   ID   B   C          D E       A ID_unit      Prob Stratum
7   7 107 207 AvSal=75 M   Delhi       1 0.1666667       1
48 48 148 248 AvSal=75 M   Delhi      11 0.1666667       1
11 11 111 211 AvSal=75 F  Kanpur      16 0.1666667       2
36 36 136 236 AvSal=75 M  Kanpur      21 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
33 33 133 233 AvSal=75 F Lucknow      36 0.1538462       4
44 44 144 244 AvSal=75 F Lucknow      40 0.1538462       4
18 18 118 218 AvSal=75 M  Mumbai      46 0.1111111       5
```

Now, suppose we aim to obtain 10 percent sample from all the variable A groups in the data frame dat1. This data framework what I have created here, this has been stored under the name dat1, data 1 ok.

So, now from this population now, I try to take here a sample and I am using here this variable here A, which is the name of the cities as my grouping variable. That means, I want to select the sample on the basis of this here A and the corresponding values will be chosen accordingly. What does this mean? For example, if you try to see here if suppose if this value is collected, say Delhi then all these corresponding values will also be selected in the sample.

Actually, that is the advantage of using this stratified function, that it gives you a lots of liberty. And so when I am saying that I want to have 10 percent sample that is proportional allocation; that means, I have to use here the numbers of size between 0 and 1 followed by a decimal. So, 10 percent can be represented as 0.1.

So, my command becomes here stratified, then the name of the data frame, then the name of the variable with respect to which we want to sample. And the size of the sample in terms of proportional allocation you have to give the number between 0 and 1 ok. So, you can see here this is what I get, when I try to execute it on the R console, right.

So, you can see here this is my here variable A and so these are the sample sizes which have been collected. So, there are 2 samples from the Delhi and 1, 2 sample from the Kanpur, 1 from Calcutta, 2 from Lucknow and 1 from Mumbai, right. So, remember that A also has been chosen randomly. So, next time if you try to repeat it, this column will not remain the same possibly.

And once you try to look at this structure, now there are 2 Delhi and there are more number of Delhi in the data frame. So, now this row is giving the other part of information, that this Delhi is coming from the ID number 7; that means, here we have here ID number here 7, if you try to see and this unit is coming in my sample.

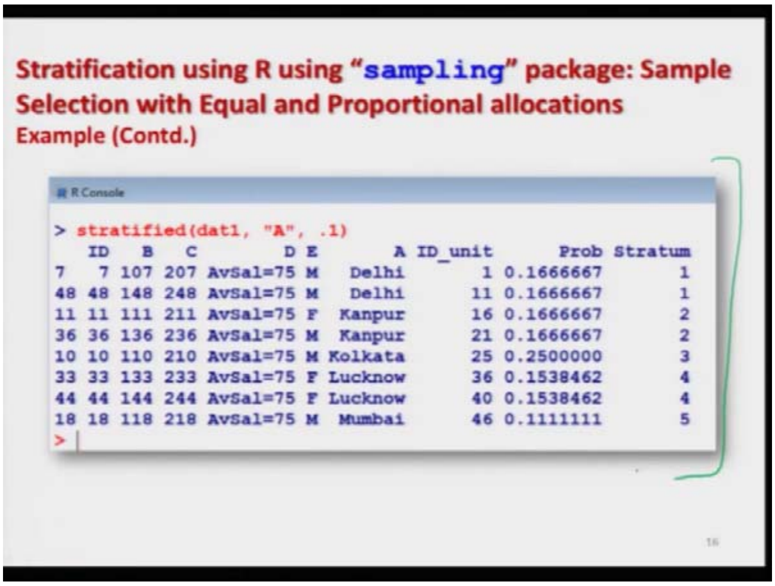
And similarly the ID corresponding to 48, 11, 36, 10 etc., they are coming in my sample. And the corresponding values of the variable B are listed here, corresponding value of C are listed here. And the corresponding value of D are listed here and corresponding value of E are listed here, right. And here it is giving you the probability of different collection and it is here mentioning that to which stratum these units belong to.

So, you can see here I have taken here 5 cities Delhi, Kanpur, Kolkata, Mumbai and Lucknow. And so this program has automatically created the strata also. It has given Delhi stratum number 1, Kanpur stratum number 2. This is clear from here, if you try to see ok, I will use a different pen. So, if you try to see A 2 this is here Kanpur 2.

So, you can see here it has given the Kanpur's means all the Kanpur names have been grouped in the stratum of number 2. And similarly here all the cities under the name Delhi they have been grouped under the stratum number 1. Kolkata has been classified into stratum number 3, Lucknow has been classified into stratum number here 4 and Mumbai has been classified into stratum number 5, right.

So, and this is our sample. Now, this is a sort of data frame and if you want to extract any variable from there, you know how to extract the information from the data frame for a given variable.

(Refer Slide Time: 20:32)



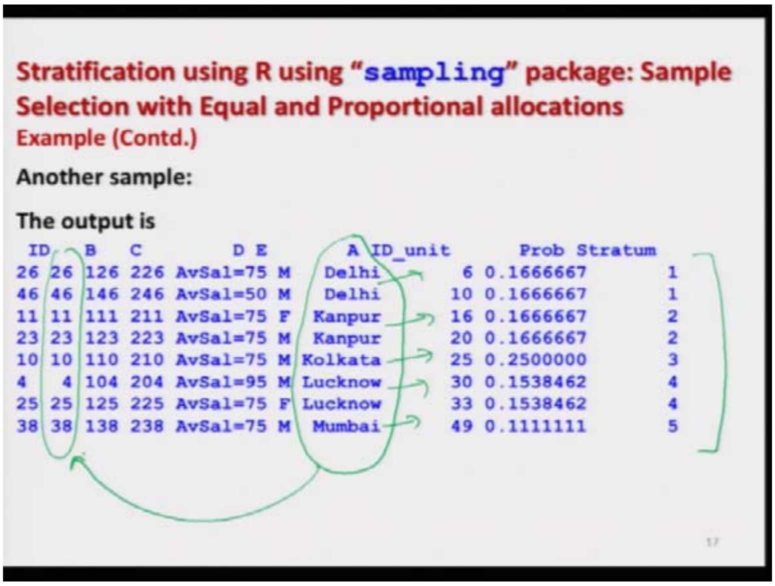
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

```
> stratified(dat1, "A", .1)
   ID   B   C          D E       A ID_unit      Prob Stratum
7   7 107 207 AvSal=75 M   Delhi       1 0.1666667       1
48 48 148 248 AvSal=75 M   Delhi      11 0.1666667       1
11 11 111 211 AvSal=75 F  Kanpur      16 0.1666667       2
36 36 136 236 AvSal=75 M  Kanpur      21 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
33 33 133 233 AvSal=75 F Lucknow      36 0.1538462       4
44 44 144 244 AvSal=75 F Lucknow      40 0.1538462       4
18 18 118 218 AvSal=75 M  Mumbai      46 0.1111111       5
>
```

And you can see here this is the screenshot what I shown you there, right.

(Refer Slide Time: 20:43)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

Another sample:

The output is

```
 ID  B   C          D E       A ID_unit      Prob Stratum
26 26 126 226 AvSal=75 M   Delhi       6 0.1666667       1
46 46 146 246 AvSal=50 M   Delhi      10 0.1666667       1
11 11 111 211 AvSal=75 F  Kanpur      16 0.1666667       2
23 23 123 223 AvSal=75 M  Kanpur      20 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
4   4 104 204 AvSal=95 M Lucknow      30 0.1538462       4
25 25 125 225 AvSal=75 F Lucknow      33 0.1538462       4
38 38 138 238 AvSal=75 M  Mumbai      49 0.1111111       5
```

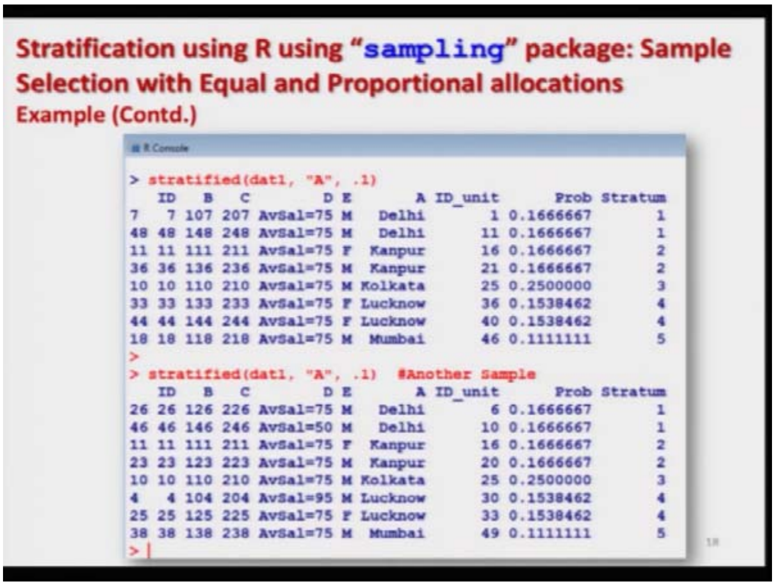
Now, if you try to take one more sample. So, you can see here I try to repeat the same command that I want to obtain 10 percent sample. You can see here now I have one more sample, but this sample is entirely different than the other sample, and the interpretation of all these values it is different.

For example, again here you can see here we have obtained the sample with respect to the name of the cities. So, this is different than the sample number one you can see here, the sample number one was Delhi, Delhi, Kanpur, Kanpur and you see here it is something like this. But the cities are the same, but they are but their IDs are now different.

You can see here right; these are means in the first sample you have got the 7th unit, 48th unit, 11th unit, 36th unit and so on. And in this sample you have got say this another units over here 26th unit, 46th unit, 11th unit. So, this is what I meant that, it is again trying to draw the 10 percent cities from Delhi, 10 percent cities from Kanpur, 10 percent from Kolkata, Lucknow as well as Mumbai.

But so these names of the city remain the same, because they are the 10 percent of the total population, but there. But now these are the different sampling units ok.

(Refer Slide Time: 22:11)



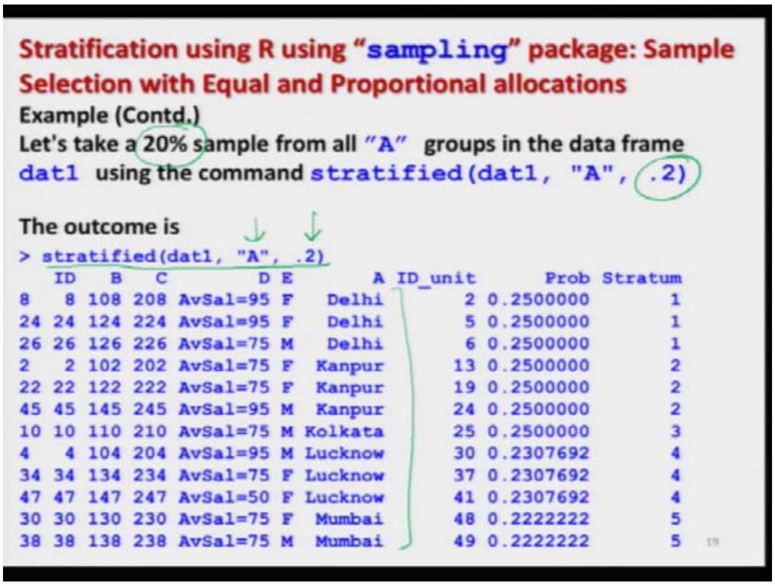
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

```
> stratified(dat1, "A", .1)
   ID   B   C        D E       A ID_unit      Prob Stratum
7   7 107 207 AvSal=75 M   Delhi       1 0.1666667       1
48 48 148 248 AvSal=75 M   Delhi      11 0.1666667       1
11 11 111 211 AvSal=75 F  Kanpur      16 0.1666667       2
36 36 136 236 AvSal=75 M  Kanpur      21 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
33 33 133 233 AvSal=75 F Lucknow      36 0.1538462       4
44 44 144 244 AvSal=75 F Lucknow      40 0.1538462       4
18 18 118 218 AvSal=75 M  Mumbai      46 0.1111111       5
>
> stratified(dat1, "A", .1)  #Another Sample
   ID   B   C        D E       A ID_unit      Prob Stratum
26 26 126 226 AvSal=75 M   Delhi       6 0.1666667       1
46 46 146 246 AvSal=50 M   Delhi      10 0.1666667       1
11 11 111 211 AvSal=75 F  Kanpur      16 0.1666667       2
23 23 123 223 AvSal=75 M  Kanpur      20 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
4   4 104 204 AvSal=95 M Lucknow      30 0.1538462       4
25 25 125 225 AvSal=75 F Lucknow      33 0.1538462       4
38 38 138 238 AvSal=75 M  Mumbai      49 0.1111111       5
> |
```

So, quickest is here means, I am just I have just given you the screenshot of both the outcomes together. So that you can see very clearly here that, here the selected IDs are this and in the second sample the selected IDs are here like this. So, these are 2 different samples, right ok.

(Refer Slide Time: 22:32)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

Let's take a 20% sample from all "A" groups in the data frame dat1 using the command stratified(dat1, "A", .2)

The outcome is

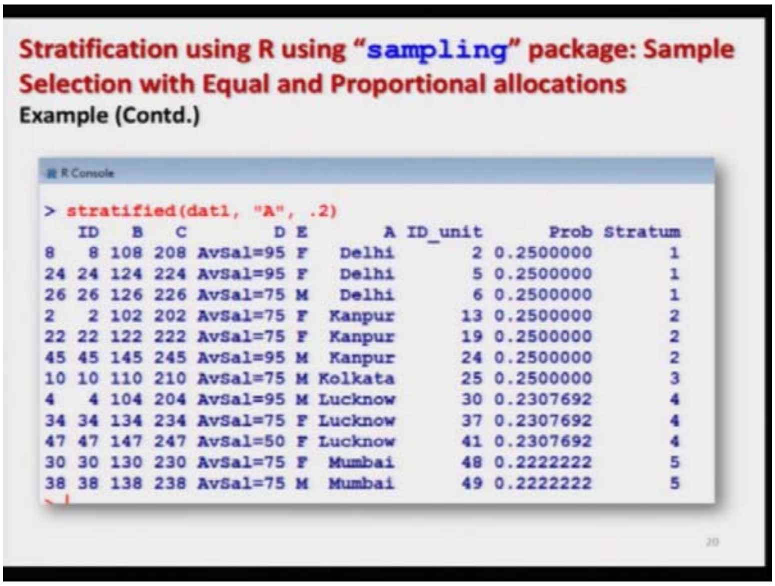
```
> stratified(dat1, "A", .2)
   ID   B   C        D E       A ID_unit      Prob Stratum
8   8 108 208 AvSal=95 F   Delhi       2 0.2500000       1
24 24 124 224 AvSal=95 F   Delhi       5 0.2500000       1
26 26 126 226 AvSal=75 M   Delhi       6 0.2500000       1
2   2 102 202 AvSal=75 F  Kanpur      13 0.2500000       2
22 22 122 222 AvSal=75 F  Kanpur      19 0.2500000       2
45 45 145 245 AvSal=95 M  Kanpur      24 0.2500000       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
4   4 104 204 AvSal=95 M Lucknow      30 0.2307692       4
34 34 134 234 AvSal=75 F Lucknow      37 0.2307692       4
47 47 147 247 AvSal=50 F Lucknow      41 0.2307692       4
30 30 130 230 AvSal=75 F  Mumbai      48 0.2222222       5
38 38 138 238 AvSal=75 M  Mumbai      49 0.2222222       5
```

19

Now, suppose if I decide that ok, instead of 10 percent, I want to obtain the 20 percent sample that is proportional allocation, but I want 20 percent of the sample, right.

So, now you can see here that, I am going to use here 20 percent, which will be indicated here 0.2. And the command remains the same here is stratified name of the data frame, name of the variable from where we want to select the samples and here the size 0.2. And you can see here, now I have got more number of samples and the interpretation goes exactly in the same way as we discussed earlier, right.

(Refer Slide Time: 23:23)



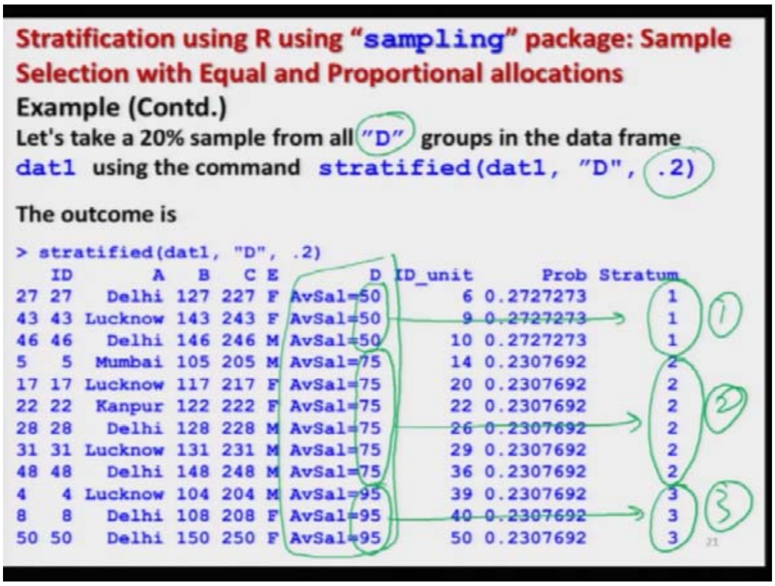
Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

```
> stratified(datl, "A", .2)
   ID   B   C         D E        A ID_unit      Prob Stratum
8   8 108 208 AvSal=95 F    Delhi       2 0.2500000       1
24 24 124 224 AvSal=95 F    Delhi       5 0.2500000       1
26 26 126 226 AvSal=75 M    Delhi       6 0.2500000       1
2   2 102 202 AvSal=75 F   Kanpur      13 0.2500000       2
22 22 122 222 AvSal=75 F   Kanpur      19 0.2500000       2
45 45 145 245 AvSal=95 M   Kanpur      24 0.2500000       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
4   4 104 204 AvSal=95 M Lucknow      30 0.2307692       4
34 34 134 234 AvSal=75 F Lucknow      37 0.2307692       4
47 47 147 247 AvSal=50 F Lucknow      41 0.2307692       4
30 30 130 230 AvSal=75 F   Mumbai      48 0.2222222       5
38 38 138 238 AvSal=75 M   Mumbai      49 0.2222222       5
>
```

And this is the screenshot of the same outcome. So, it is very simple and straight forward.

(Refer Slide Time: 23:27)



Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

Let's take a 20% sample from all "D" groups in the data frame datl using the command stratified(datl, "D", .2)

The outcome is

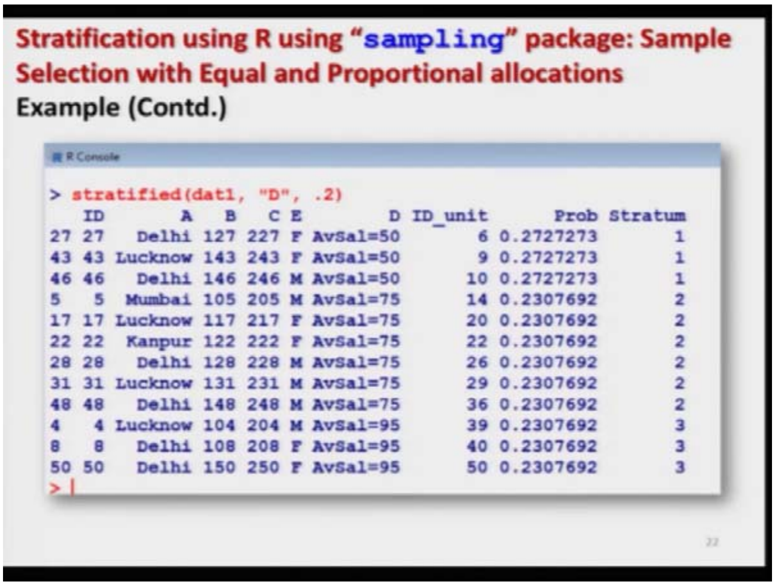
```
> stratified(datl, "D", .2)
   ID       A   B   C E        D ID_unit      Prob Stratum
27 27   Delhi 127 227 F AvSal=50       6 0.2727273       1
43 43 Lucknow 143 243 F AvSal=50       9 0.2727273       1
46 46   Delhi 146 246 M AvSal=50      10 0.2727273       1
5   5  Mumbai 105 205 M AvSal=75      14 0.2307692       2
17 17 Lucknow 117 217 F AvSal=75      20 0.2307692       2
22 22  Kanpur 122 222 F AvSal=75      22 0.2307692       2
28 28   Delhi 128 228 M AvSal=75      26 0.2307692       2
31 31 Lucknow 131 231 M AvSal=75      29 0.2307692       2
48 48   Delhi 148 248 M AvSal=75      36 0.2307692       2
4   4 Lucknow 104 204 M AvSal=95      39 0.2307692       3
8   8   Delhi 108 208 F AvSal=95      40 0.2307692       3
50 50   Delhi 150 250 F AvSal=95      50 0.2307692       3
```

Now, suppose I take a decision that in the earlier example, we have chosen the variable capital A with respect to which we want to select the sample. Now, suppose I want to choose another variable, suppose I choose here variable D and from there I want to obtain 20 percent sample from each of this strata, right.

So, now you can see here that this is here D and the there are now 3 values corresponding to 50, which are classified into stratum number 1. There are 6 values corresponding to average salary which are classified into stratum number 2 and there are 3 values 95 which are classified in stratum number 3. And now you can see here in this case there are only 3 stratum, why? I can show you here the way you have constructed the data frame.

Here you have taken here only 3 possible values, I will use a different pen you can see here 1, 2 and here 3. So, that is the advantage of creating a data set yourself to understand the basics. So, now since there are only 3 possible classes, total number of classes which have been reported here you can see here they are also 1, 2, and 3. So, I hope now this will make the things more clear that how the sample has been obtained.

(Refer Slide Time: 25:16)

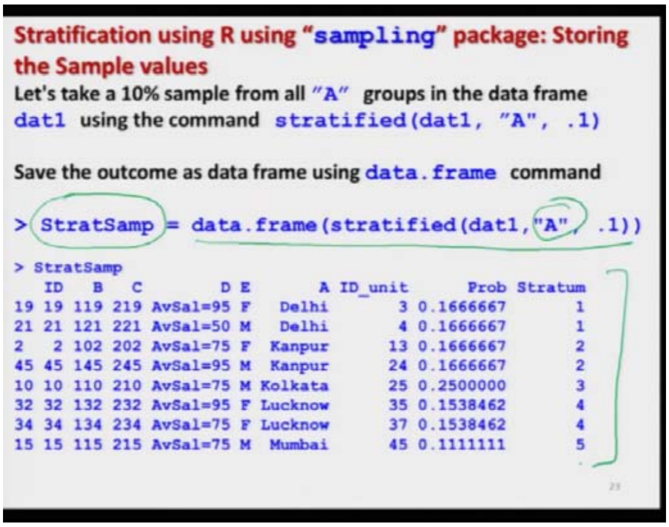


Stratification using R using "sampling" package: Sample Selection with Equal and Proportional allocations

Example (Contd.)

```
> stratified(dat1, "D", .2)
   ID       A   B   C E         D ID_unit      Prob Stratum
27 27   Delhi 127 227 F AvSal=50       6 0.2727273       1
43 43 Lucknow 143 243 F AvSal=50       9 0.2727273       1
46 46   Delhi 146 246 M AvSal=50      10 0.2727273       1
5   5  Mumbai 105 205 M AvSal=75      14 0.2307692       2
17 17 Lucknow 117 217 F AvSal=75      20 0.2307692       2
22 22  Kanpur 122 222 F AvSal=75      22 0.2307692       2
28 28   Delhi 128 228 M AvSal=75      26 0.2307692       2
31 31 Lucknow 131 231 M AvSal=75      29 0.2307692       2
48 48   Delhi 148 248 M AvSal=75      36 0.2307692       2
4   4 Lucknow 104 204 M AvSal=95      39 0.2307692       3
8   8   Delhi 108 208 F AvSal=95      40 0.2307692       3
50 50   Delhi 150 250 F AvSal=95      50 0.2307692       3
> |
```

(Refer Slide Time: 25:21)



Stratification using R using "sampling" package: Storing the Sample values

Let's take a 10% sample from all "A" groups in the data frame dat1 using the command stratified(dat1, "A", .1)

Save the outcome as data frame using data.frame command

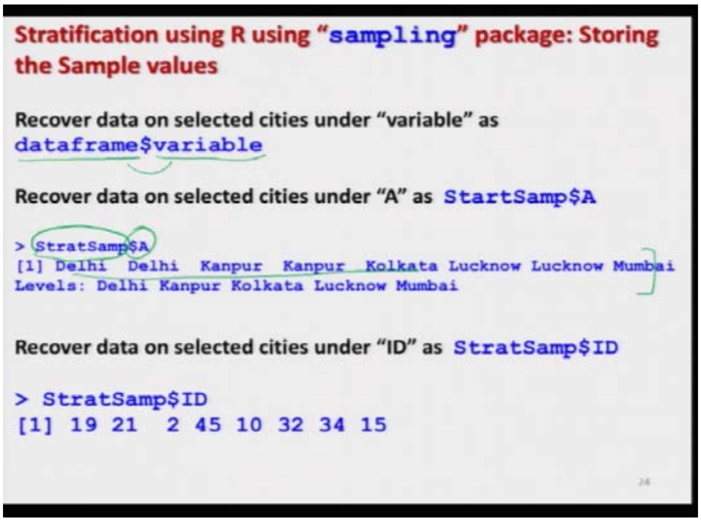
```
> StratSamp = data.frame(stratified(dat1,"A", .1))
```

```
> StratSamp
   ID   B   C        D E       A ID_unit      Prob Stratum
19 19 119 219 AvSal=95 F   Delhi       3 0.1666667       1
21 21 121 221 AvSal=50 M   Delhi       4 0.1666667       1
2   2 102 202 AvSal=75 F  Kanpur      13 0.1666667       2
45 45 145 245 AvSal=95 M  Kanpur      24 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
32 32 132 232 AvSal=95 F Lucknow      35 0.1538462       4
34 34 134 234 AvSal=75 F Lucknow      37 0.1538462       4
15 15 115 215 AvSal=75 M  Mumbai      45 0.1111111       5
```

And this is here the screenshot of the same outcome, right. Now, what I have done that I will simply saved the output of this command. So, that I can use it further to show you more thing, because every time if I try to execute it, then possibly it will give me a different sample and then it will be difficult for me to explain you.

Suppose, I execute the same command and whatever is my outcome in which I am trying to obtain 10 percent of the sample with respect to the variable A. And I am whatever is my outcome I am trying to store it in a variable say a StratSamp, that means stratified sample. And this is the outcome which I have got here; as I said when I try to do it on the R console this is going to be difficult.

(Refer Slide Time: 26:06)



Stratification using R using "sampling" package: Storing the Sample values

Recover data on selected cities under "variable" as dataframe$variable

Recover data on selected cities under "A" as StartSamp$A

```
> StratSamp$A
[1] Delhi  Delhi  Kanpur  Kanpur  Kolkata Lucknow Lucknow Mumbai
Levels: Delhi Kanpur Kolkata Lucknow Mumbai
```

Recover data on selected cities under "ID" as StratSamp$ID

```
> StratSamp$ID
[1] 19 21  2 45 10 32 34 15
```

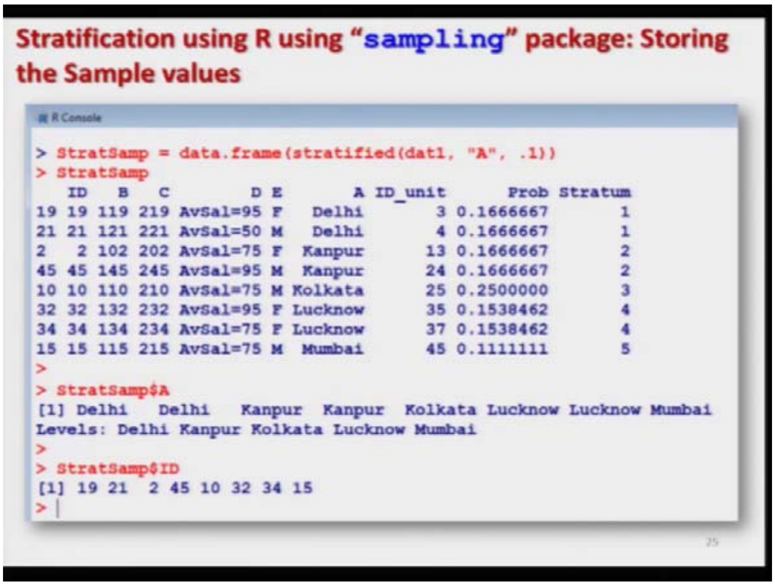
Now, if you want to recover the data on a particular variable from this data frame, then you know that the rule is very very simple, you simply have to write down the name of the data frame and followed by the variable and both are going to be joined by dollar sign.

Suppose, if I, if suppose I have got this data set and suppose I want to recover that, what are the cities which are selected here, right. So, you can simply use here say, means the name of the data frame StratSamp and dollar. The name of variable capital A and you can see here this is the data set which has been obtained you can see Delhi Delhi, Kanpur Kanpur, Kolkata etc. and you can see here Delhi Delhi Kanpur Kanpur Kolkata etc.

So, that this data set has been obtained there. Similarly, if you want to obtain or recover the data set on the selected cities under the variable name ID. So, I simply have to give here the name of the data frame followed by ID and joined by a variable name which is ID. So, you can see here you have you are getting here the outcome 19, 21, 2, 45etc.

So, you can see here this is the same 19, 21, 2, 45 etc. So, this data has been recovered by using this command ok.

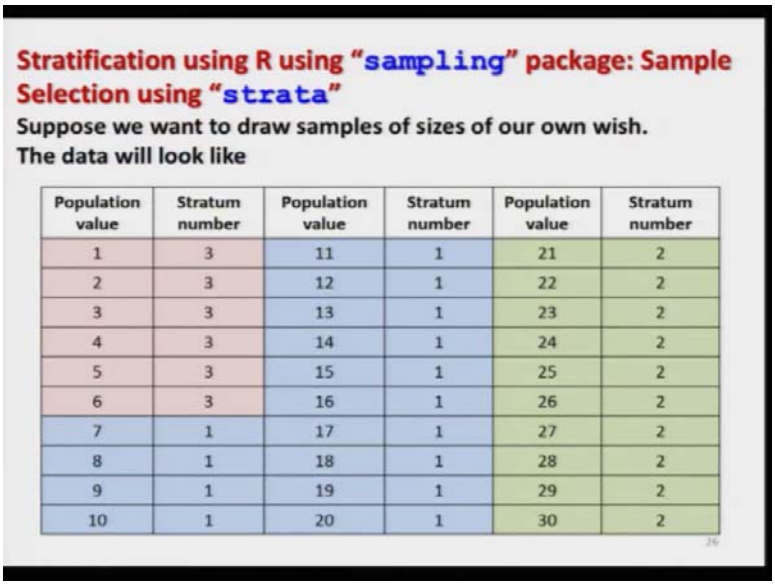
(Refer Slide Time: 27:28)



Stratification using R using "sampling" package: Storing the Sample values

```
> StratSamp = data.frame(stratified(dat1, "A", .1))
> StratSamp
   ID   B   C        D E       A ID_unit      Prob Stratum
19 19 119 219 AvSal=95 F   Delhi       3 0.1666667       1
21 21 121 221 AvSal=50 M   Delhi       4 0.1666667       1
2   2 102 202 AvSal=75 F  Kanpur      13 0.1666667       2
45 45 145 245 AvSal=95 M  Kanpur      24 0.1666667       2
10 10 110 210 AvSal=75 M Kolkata      25 0.2500000       3
32 32 132 232 AvSal=95 F Lucknow      35 0.1538462       4
34 34 134 234 AvSal=75 F Lucknow      37 0.1538462       4
15 15 115 215 AvSal=75 M  Mumbai      45 0.1111111       5
>
> StratSamp$A
[1] Delhi   Delhi   Kanpur  Kanpur  Kolkata Lucknow Lucknow Mumbai
Levels: Delhi Kanpur Kolkata Lucknow Mumbai
>
> StratSamp$ID
[1] 19 21  2 45 10 32 34 15
> |
```

(Refer Slide Time: 27:34)



Stratification using R using "sampling" package: Sample Selection using "strata"

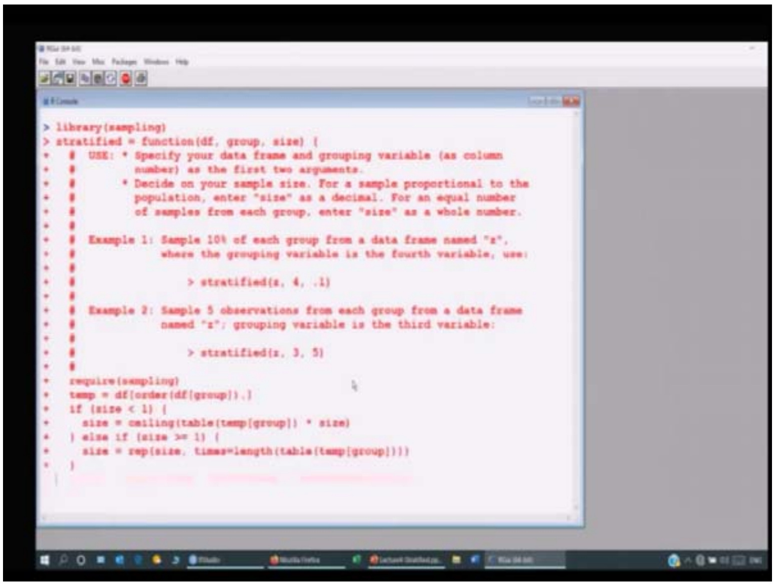
Suppose we want to draw samples of sizes of our own wish.
The data will look like

Population value	Stratum number	Population value	Stratum number	Population value	Stratum number
1	3	11	1	21	2
2	3	12	1	22	2
3	3	13	1	23	2
4	3	14	1	24	2
5	3	15	1	25	2
6	3	16	1	26	2
7	1	17	1	27	2
8	1	18	1	28	2
9	1	19	1	29	2
10	1	20	1	30	2

26

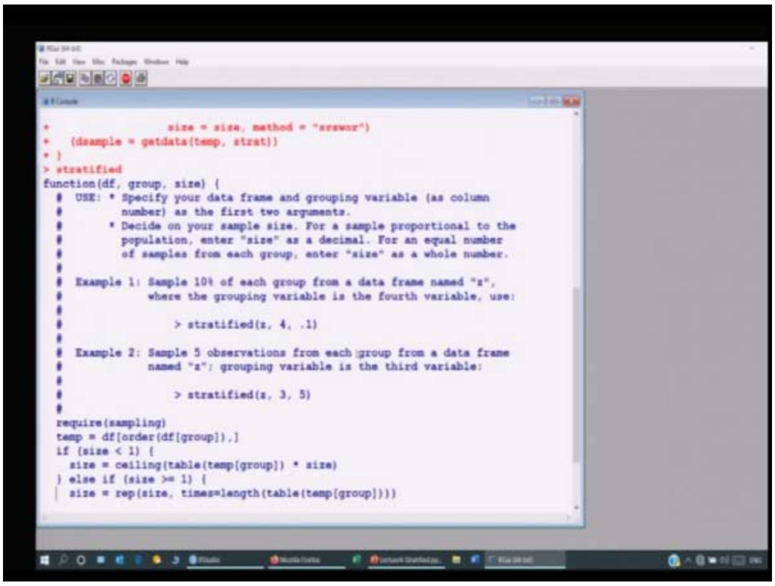
And this is here the outcome ok. I have shown you that how you can use this script stratify.R to draw the sample. Now, I would try to first show you it on the R console. And then I will try to show you that, how you can use the another package here, strata to select the things, right. So, let us come to the R console and first I try to load the sampling package, yes I already have installed it on my computer.

(Refer Slide Time: 28:12)



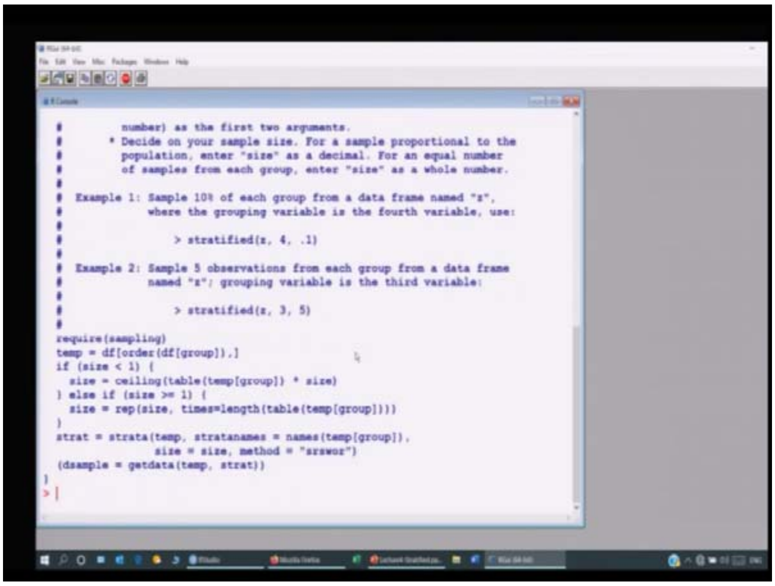
```
> library(sampling)
> stratified = function(df, group, size) {
+   #  USE: * Specify your data frame and grouping variable (as column
+   #          number) as the first two arguments.
+   #        * Decide on your sample size. For a sample proportional to the
+   #          population, enter "size" as a decimal. For an equal number
+   #          of samples from each group, enter "size" as a whole number.
+   #
+   #  Example 1: Sample 10% of each group from a data frame named "z",
+   #             where the grouping variable is the fourth variable, use:
+   #
+   #                 > stratified(z, 4, .1)
+   #
+   #  Example 2: Sample 5 observations from each group from a data frame
+   #             named "z"; grouping variable is the third variable:
+   #
+   #                 > stratified(z, 3, 5)
+   #
+   require(sampling)
+   temp = df[order(df[group]),]
+   if (size < 1) {
+     size = ceiling(table(temp[group]) * size)
+   } else if (size >= 1) {
+     size = rep(size, times=length(table(temp[group])))
+   }
```

(Refer Slide Time: 28:18)



```
+                  size = size, method = "srswor")
+   (dsample = getdata(temp, strat))
+ }
> stratified
function(df, group, size) {
  #  USE: * Specify your data frame and grouping variable (as column
  #         number) as the first two arguments.
  #       * Decide on your sample size. For a sample proportional to the
  #         population, enter "size" as a decimal. For an equal number
  #         of samples from each group, enter "size" as a whole number.
  #
  #  Example 1: Sample 10% of each group from a data frame named "z",
  #             where the grouping variable is the fourth variable, use:
  #
  #                 > stratified(z, 4, .1)
  #
  #  Example 2: Sample 5 observations from each group from a data frame
  #             named "z"; grouping variable is the third variable:
  #
  #                 > stratified(z, 3, 5)
  #
  require(sampling)
  temp = df[order(df[group]),]
  if (size < 1) {
    size = ceiling(table(temp[group]) * size)
  } else if (size >= 1) {
    size = rep(size, times=length(table(temp[group])))
```

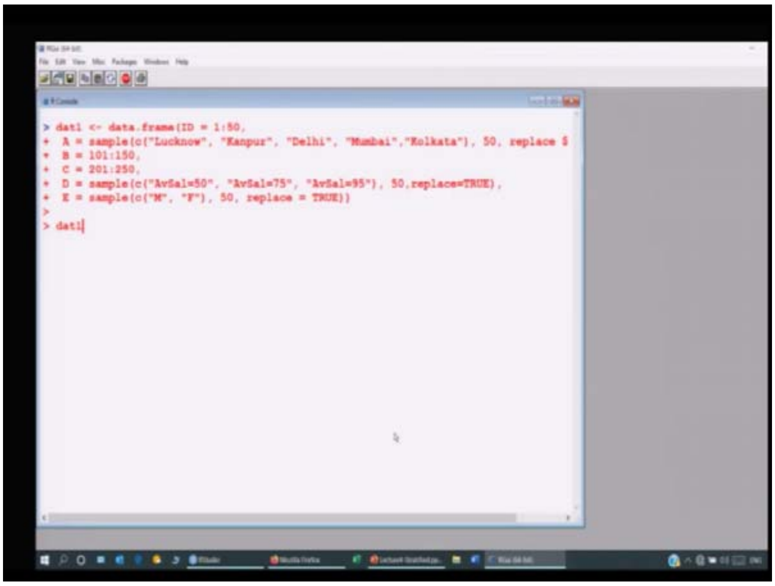
(Refer Slide Time: 28:25)



```
  #         number) as the first two arguments.
  #       * Decide on your sample size. For a sample proportional to the
  #         population, enter "size" as a decimal. For an equal number
  #         of samples from each group, enter "size" as a whole number.
  #
  #  Example 1: Sample 10% of each group from a data frame named "z",
  #             where the grouping variable is the fourth variable, use:
  #
  #                 > stratified(z, 4, .1)
  #
  #  Example 2: Sample 5 observations from each group from a data frame
  #             named "z"; grouping variable is the third variable:
  #
  #                 > stratified(z, 3, 5)
  #
  require(sampling)
  temp = df[order(df[group]),]
  if (size < 1) {
    size = ceiling(table(temp[group]) * size)
  } else if (size >= 1) {
    size = rep(size, times=length(table(temp[group])))
  }
  strat = strata(temp, stratanames = names(temp[group]),
                 size = size, method = "srswor")
  (dsample = getdata(temp, strat))
}
> |
```

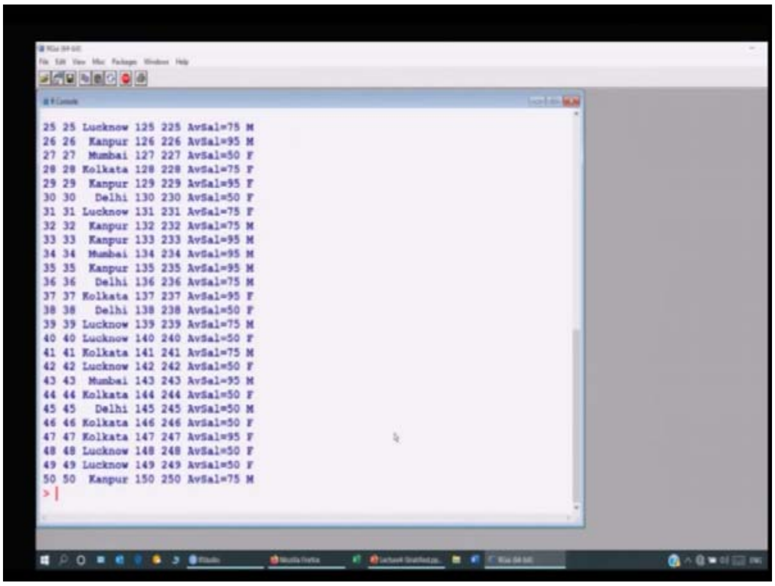
So, and now I try to copy this program stratify.R and I simply try to here paste and you can see here. Now, this program is on my laptop, you can see here stratified, you can see here this is the program which I was saying ok, right ok. So, now let me clear the screen and we come back to our slide and I try to create the same data frame from where we have obtained the sample.

(Refer Slide Time: 28:43)



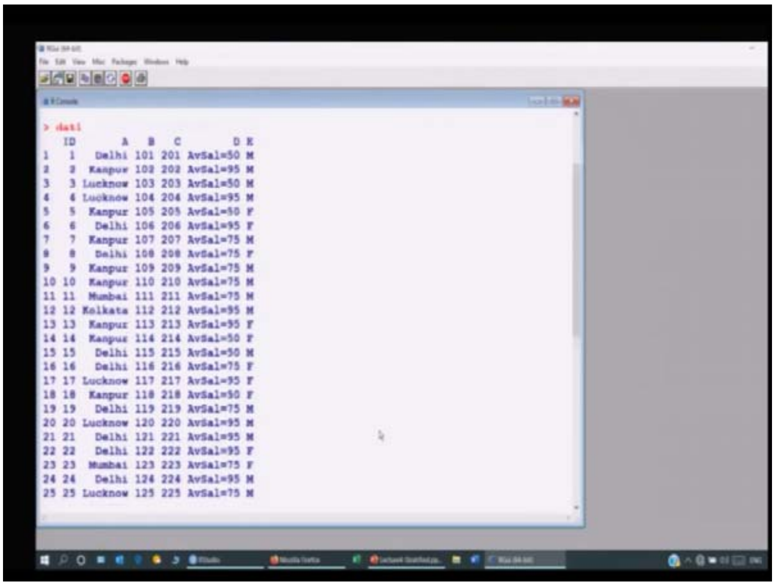
```
> dat1 <- data.frame(ID = 1:50,
+  A = sample(c("Lucknow", "Kanpur", "Delhi", "Mumbai","Kolkata"), 50, replace $
+  B = 101:150,
+  C = 201:250,
+  D = sample(c("AvSal=50", "AvSal=75", "AvSal=95"), 50,replace=TRUE),
+  E = sample(c("M", "F"), 50, replace = TRUE))
>
> dat1
```

(Refer Slide Time: 28:46)



```
25 25 Lucknow 125 225 AvSal=75 M
26 26  Kanpur 126 226 AvSal=95 M
27 27  Mumbai 127 227 AvSal=50 F
28 28 Kolkata 128 228 AvSal=75 F
29 29  Kanpur 129 229 AvSal=95 F
30 30   Delhi 130 230 AvSal=50 F
31 31 Lucknow 131 231 AvSal=75 F
32 32  Kanpur 132 232 AvSal=75 M
33 33  Kanpur 133 233 AvSal=95 M
34 34  Mumbai 134 234 AvSal=95 M
35 35  Kanpur 135 235 AvSal=95 M
36 36   Delhi 136 236 AvSal=75 M
37 37 Kolkata 137 237 AvSal=95 F
38 38   Delhi 138 238 AvSal=50 F
39 39 Lucknow 139 239 AvSal=75 M
40 40 Lucknow 140 240 AvSal=50 F
41 41 Kolkata 141 241 AvSal=75 M
42 42 Lucknow 142 242 AvSal=50 F
43 43  Mumbai 143 243 AvSal=95 M
44 44 Kolkata 144 244 AvSal=50 F
45 45   Delhi 145 245 AvSal=50 M
46 46 Kolkata 146 246 AvSal=50 F
47 47 Kolkata 147 247 AvSal=95 F
48 48 Lucknow 148 248 AvSal=50 F
49 49 Lucknow 149 249 AvSal=50 F
50 50  Kanpur 150 250 AvSal=75 M
>
```

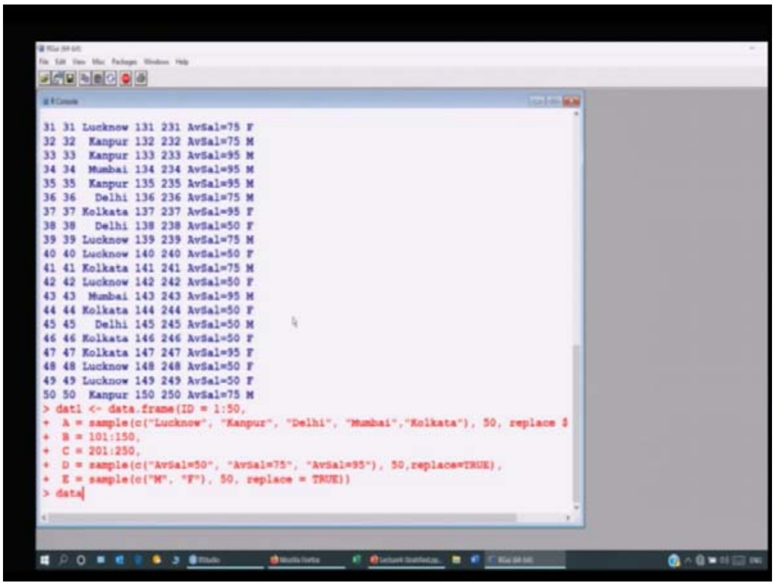
(Refer Slide Time: 28:51)



```
> dat1
   ID       A   B   C        D E
1   1   Delhi 101 201 AvSal=50 M
2   2  Kanpur 102 202 AvSal=95 M
3   3 Lucknow 103 203 AvSal=50 M
4   4 Lucknow 104 204 AvSal=95 M
5   5  Kanpur 105 205 AvSal=50 F
6   6   Delhi 106 206 AvSal=95 F
7   7  Kanpur 107 207 AvSal=75 M
8   8   Delhi 108 208 AvSal=75 F
9   9  Kanpur 109 209 AvSal=75 M
10 10  Kanpur 110 210 AvSal=75 M
11 11  Mumbai 111 211 AvSal=75 M
12 12 Kolkata 112 212 AvSal=95 M
13 13  Kanpur 113 213 AvSal=95 F
14 14  Kanpur 114 214 AvSal=50 F
15 15   Delhi 115 215 AvSal=50 M
16 16   Delhi 116 216 AvSal=75 F
17 17 Lucknow 117 217 AvSal=95 F
18 18  Kanpur 118 218 AvSal=50 F
19 19   Delhi 119 219 AvSal=75 M
20 20 Lucknow 120 220 AvSal=95 M
21 21   Delhi 121 221 AvSal=95 M
22 22   Delhi 122 222 AvSal=95 F
23 23  Mumbai 123 223 AvSal=75 F
24 24   Delhi 124 224 AvSal=95 M
25 25 Lucknow 125 225 AvSal=75 M
```

So, you can see here this is my data frame and you can see here this is how it will look like, right. These are here first column is here ID then A, B, C, D, E and you can see here this will look like this. So and means if means since I am using here the commands like sample etc.

(Refer Slide Time: 29:11)

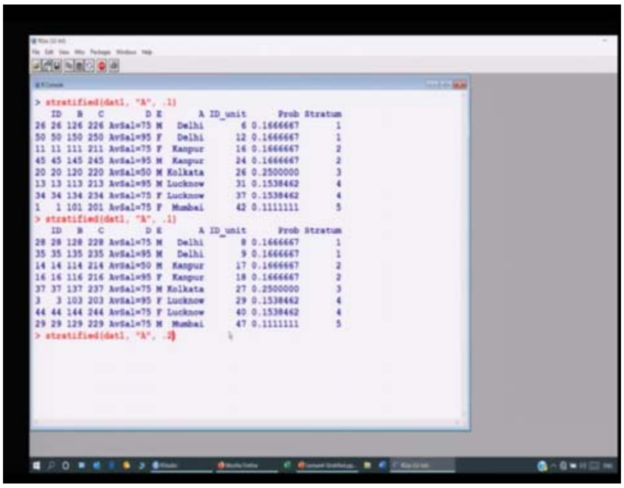


```
31 31 Lucknow 131 231 AvSal=75 F
32 32  Kanpur 132 232 AvSal=75 M
33 33  Kanpur 133 233 AvSal=95 M
34 34  Mumbai 134 234 AvSal=95 M
35 35  Kanpur 135 235 AvSal=95 M
36 36   Delhi 136 236 AvSal=75 M
37 37 Kolkata 137 237 AvSal=95 F
38 38   Delhi 138 238 AvSal=50 F
39 39 Lucknow 139 239 AvSal=75 M
40 40 Lucknow 140 240 AvSal=50 F
41 41 Kolkata 141 241 AvSal=75 M
42 42 Lucknow 142 242 AvSal=50 F
43 43  Mumbai 143 243 AvSal=95 M
44 44 Kolkata 144 244 AvSal=50 F
45 45   Delhi 145 245 AvSal=50 M
46 46 Kolkata 146 246 AvSal=50 F
47 47 Kolkata 147 247 AvSal=95 F
48 48 Lucknow 148 248 AvSal=50 F
49 49 Lucknow 149 249 AvSal=50 F
50 50  Kanpur 150 250 AvSal=75 M
> dat1 <- data.frame(ID = 1:50,
+  A = sample(c("Lucknow", "Kanpur", "Delhi", "Mumbai","Kolkata"), 50, replace $
+  B = 101:150,
+  C = 201:250,
+  D = sample(c("AvSal=50", "AvSal=75", "AvSal=95"), 50,replace=TRUE),
+  E = sample(c("M", "F"), 50, replace = TRUE))
> data
```

So, if I try to repeat this command here, once again to create this another data set. Now, this data set will look different than the earlier one. For example, you can see here the first 3 units are a Kanpur, Mumbai, Delhi and in the; and in the first case, it was Delhi, Kanpur, Lucknow.

So, that is what I said that every time if I try to change it I will get a different outcome. So, but now let us fix this one and I try to draw here. The samples using the command here stratified function.

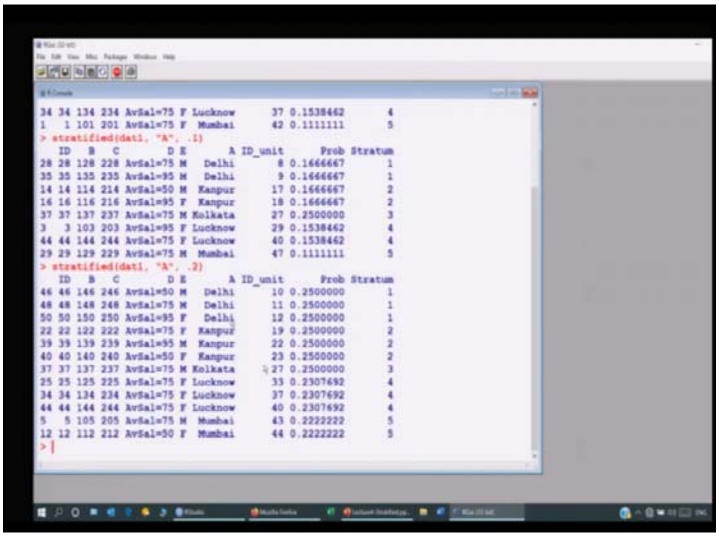
(Refer Slide Time: 29:45)



```
> stratified(dat1, "A", .1)
   ID   B   C        D E       A ID_unit      Prob Stratum
26 26 126 226 AvSal=75 M   Delhi       6 0.1666667       1
50 50 150 250 AvSal=95 F   Delhi      12 0.1666667       1
11 11 111 211 AvSal=75 F  Kanpur      16 0.1666667       2
45 45 145 245 AvSal=95 M  Kanpur      24 0.1666667       2
20 20 120 220 AvSal=50 M Kolkata      26 0.2500000       3
13 13 113 213 AvSal=95 M Lucknow      31 0.1538462       4
34 34 134 234 AvSal=75 F Lucknow      37 0.1538462       4
1   1 101 201 AvSal=75 F  Mumbai      42 0.1111111       5
> stratified(dat1, "A", .1)
   ID   B   C        D E       A ID_unit      Prob Stratum
28 28 128 228 AvSal=75 M   Delhi       8 0.1666667       1
35 35 135 235 AvSal=95 M   Delhi       9 0.1666667       1
14 14 114 214 AvSal=50 M  Kanpur      17 0.1666667       2
16 16 116 216 AvSal=95 F  Kanpur      18 0.1666667       2
37 37 137 237 AvSal=75 M Kolkata      27 0.2500000       3
3   3 103 203 AvSal=95 F Lucknow      29 0.1538462       4
44 44 144 244 AvSal=75 F Lucknow      40 0.1538462       4
29 29 129 229 AvSal=75 M  Mumbai      47 0.1111111       5
> stratified(dat1, "A", .2)
```

So, now I clear the screen and I put here this command stratified. So, you can see here you have obtained this data set and if you try to repeat this command, you will see here another stratified sample. But which is different than other the first one. For example, here in the first sample the ID units are 6, 12, 16; and in other cases these are 8, 9 and 17, right.

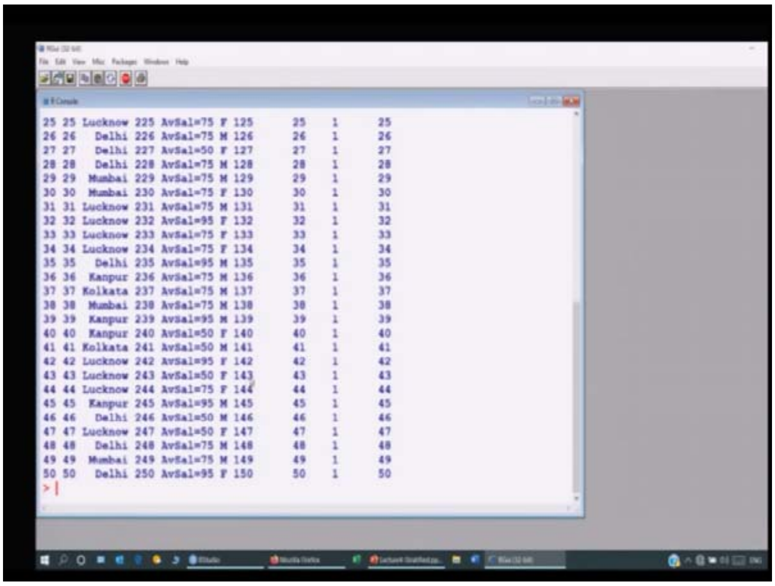
(Refer Slide Time: 30:18)



```
34 34 134 234 AvSal=75 F Lucknow      37 0.1538462       4
1   1 101 201 AvSal=75 F  Mumbai      42 0.1111111       5
> stratified(dat1, "A", .1)
   ID   B   C        D E       A ID_unit      Prob Stratum
28 28 128 228 AvSal=75 M   Delhi       8 0.1666667       1
35 35 135 235 AvSal=95 M   Delhi       9 0.1666667       1
14 14 114 214 AvSal=50 M  Kanpur      17 0.1666667       2
16 16 116 216 AvSal=95 F  Kanpur      18 0.1666667       2
37 37 137 237 AvSal=75 M Kolkata      27 0.2500000       3
3   3 103 203 AvSal=95 F Lucknow      29 0.1538462       4
44 44 144 244 AvSal=75 F Lucknow      40 0.1538462       4
29 29 129 229 AvSal=75 M  Mumbai      47 0.1111111       5
> stratified(dat1, "A", .2)
   ID   B   C        D E       A ID_unit      Prob Stratum
46 46 146 246 AvSal=50 M   Delhi      10 0.2500000       1
48 48 148 248 AvSal=75 M   Delhi      11 0.2500000       1
50 50 150 250 AvSal=95 F   Delhi      12 0.2500000       1
22 22 122 222 AvSal=75 F  Kanpur      19 0.2500000       2
39 39 139 239 AvSal=95 M  Kanpur      22 0.2500000       2
40 40 140 240 AvSal=50 F  Kanpur      23 0.2500000       2
37 37 137 237 AvSal=75 M Kolkata      27 0.2500000       3
25 25 125 225 AvSal=75 F Lucknow      33 0.2307692       4
34 34 134 234 AvSal=75 F Lucknow      37 0.2307692       4
44 44 144 244 AvSal=75 F Lucknow      40 0.2307692       4
5   5 105 205 AvSal=75 M  Mumbai      43 0.2222222       5
12 12 112 212 AvSal=50 F  Mumbai      44 0.2222222       5
> |
```

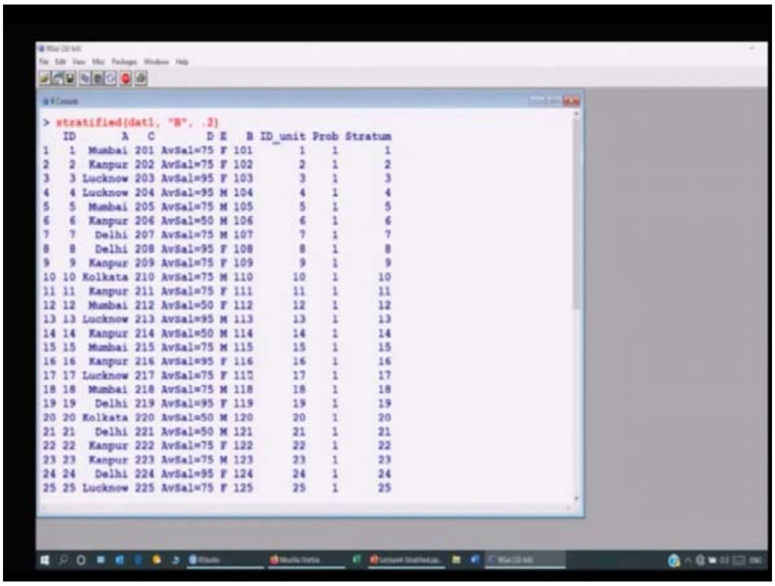
And similarly if you want to take here us stratified sample, which is 20 percent. You can see here now this is another sample.

(Refer Slide Time: 30:30)



```
25 25 Lucknow 225 AvSal=75 F 125       25    1       25
26 26   Delhi 226 AvSal=75 M 126       26    1       26
27 27   Delhi 227 AvSal=50 F 127       27    1       27
28 28   Delhi 228 AvSal=75 M 128       28    1       28
29 29  Mumbai 229 AvSal=75 M 129       29    1       29
30 30  Mumbai 230 AvSal=75 F 130       30    1       30
31 31 Lucknow 231 AvSal=75 M 131       31    1       31
32 32 Lucknow 232 AvSal=95 F 132       32    1       32
33 33 Lucknow 233 AvSal=75 F 133       33    1       33
34 34 Lucknow 234 AvSal=75 F 134       34    1       34
35 35   Delhi 235 AvSal=95 M 135       35    1       35
36 36  Kanpur 236 AvSal=75 M 136       36    1       36
37 37 Kolkata 237 AvSal=75 M 137       37    1       37
38 38  Mumbai 238 AvSal=75 M 138       38    1       38
39 39  Kanpur 239 AvSal=95 M 139       39    1       39
40 40  Kanpur 240 AvSal=50 F 140       40    1       40
41 41 Kolkata 241 AvSal=50 M 141       41    1       41
42 42 Lucknow 242 AvSal=95 F 142       42    1       42
43 43 Lucknow 243 AvSal=50 F 143       43    1       43
44 44 Lucknow 244 AvSal=75 F 144       44    1       44
45 45  Kanpur 245 AvSal=95 M 145       45    1       45
46 46   Delhi 246 AvSal=50 M 146       46    1       46
47 47 Lucknow 247 AvSal=50 F 147       47    1       47
48 48   Delhi 248 AvSal=75 M 148       48    1       48
49 49  Mumbai 249 AvSal=75 M 149       49    1       49
50 50   Delhi 250 AvSal=95 F 150       50    1       50
> |
```

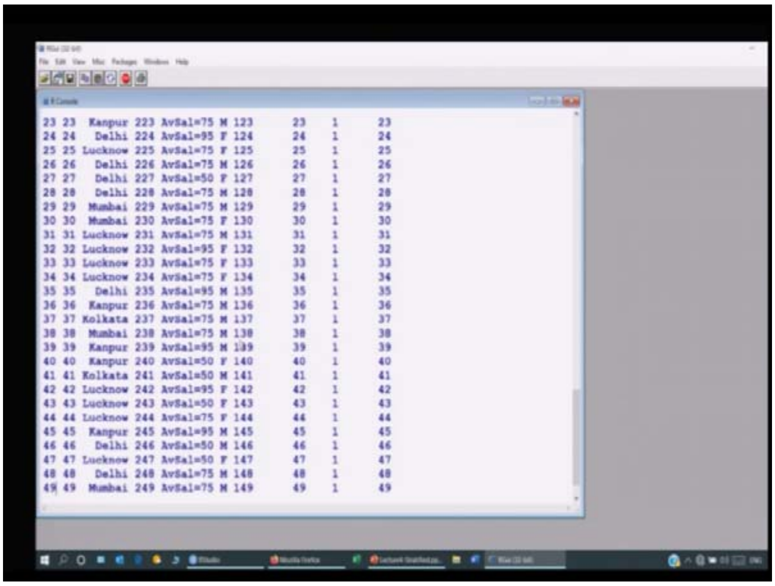
(Refer Slide Time: 30:31)



```
> stratified(dat1, "B", .2)
   ID       A   C        D E   B ID_unit Prob Stratum
1   1  Mumbai 201 AvSal=75 F 101       1    1       1
2   2  Kanpur 202 AvSal=75 F 102       2    1       2
3   3 Lucknow 203 AvSal=95 F 103       3    1       3
4   4 Lucknow 204 AvSal=95 M 104       4    1       4
5   5  Mumbai 205 AvSal=75 M 105       5    1       5
6   6  Kanpur 206 AvSal=50 M 106       6    1       6
7   7   Delhi 207 AvSal=75 M 107       7    1       7
8   8   Delhi 208 AvSal=95 F 108       8    1       8
9   9  Kanpur 209 AvSal=75 F 109       9    1       9
10 10 Kolkata 210 AvSal=75 M 110      10    1      10
11 11  Kanpur 211 AvSal=75 F 111      11    1      11
12 12  Mumbai 212 AvSal=50 F 112      12    1      12
13 13 Lucknow 213 AvSal=95 M 113      13    1      13
14 14  Kanpur 214 AvSal=50 M 114      14    1      14
15 15  Mumbai 215 AvSal=75 M 115      15    1      15
16 16  Kanpur 216 AvSal=95 F 116      16    1      16
17 17 Lucknow 217 AvSal=75 F 117      17    1      17
18 18  Mumbai 218 AvSal=75 M 118      18    1      18
19 19   Delhi 219 AvSal=95 F 119      19    1      19
20 20 Kolkata 220 AvSal=50 M 120      20    1      20
21 21   Delhi 221 AvSal=50 M 121      21    1      21
22 22  Kanpur 222 AvSal=75 F 122      22    1      22
23 23  Kanpur 223 AvSal=75 M 123      23    1      23
24 24   Delhi 224 AvSal=95 F 124      24    1      24
25 25 Lucknow 225 AvSal=75 F 125      25    1      25
```

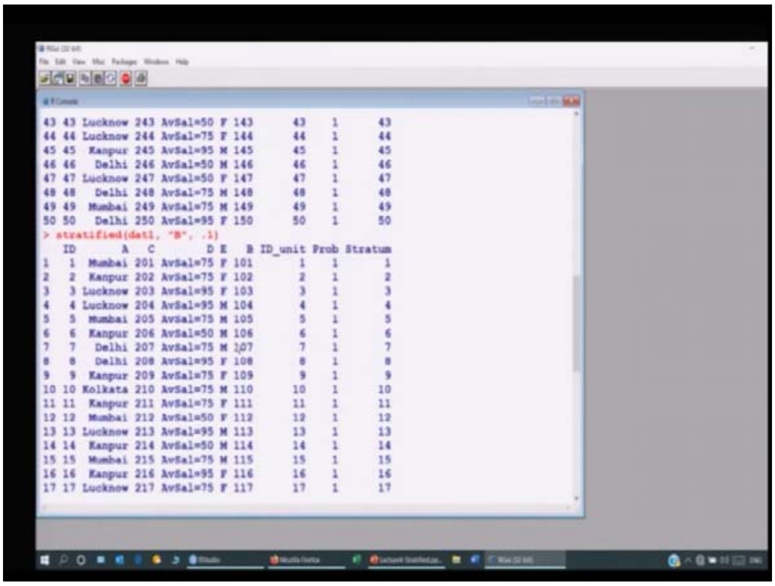
And if you want to change the variable name with which you want to select the value. Suppose I make it here B. So, you can see here this is my 20 percent sample.

(Refer Slide Time: 30:43)



```
23 23  Kanpur 223 AvSal=75 M 123       23    1       23
24 24   Delhi 224 AvSal=95 F 124       24    1       24
25 25 Lucknow 225 AvSal=75 F 125       25    1       25
26 26   Delhi 226 AvSal=75 M 126       26    1       26
27 27   Delhi 227 AvSal=50 F 127       27    1       27
28 28   Delhi 228 AvSal=75 M 128       28    1       28
29 29  Mumbai 229 AvSal=75 M 129       29    1       29
30 30  Mumbai 230 AvSal=75 F 130       30    1       30
31 31 Lucknow 231 AvSal=75 M 131       31    1       31
32 32 Lucknow 232 AvSal=95 F 132       32    1       32
33 33 Lucknow 233 AvSal=75 F 133       33    1       33
34 34 Lucknow 234 AvSal=75 F 134       34    1       34
35 35   Delhi 235 AvSal=95 M 135       35    1       35
36 36  Kanpur 236 AvSal=75 M 136       36    1       36
37 37 Kolkata 237 AvSal=75 M 137       37    1       37
38 38  Mumbai 238 AvSal=75 M 138       38    1       38
39 39  Kanpur 239 AvSal=95 M 139       39    1       39
40 40  Kanpur 240 AvSal=50 F 140       40    1       40
41 41 Kolkata 241 AvSal=50 M 141       41    1       41
42 42 Lucknow 242 AvSal=95 F 142       42    1       42
43 43 Lucknow 243 AvSal=50 F 143       43    1       43
44 44 Lucknow 244 AvSal=75 F 144       44    1       44
45 45  Kanpur 245 AvSal=95 M 145       45    1       45
46 46   Delhi 246 AvSal=50 M 146       46    1       46
47 47 Lucknow 247 AvSal=50 F 147       47    1       47
48 48   Delhi 248 AvSal=75 M 148       48    1       48
49 49  Mumbai 249 AvSal=75 M 149       49    1       49
```

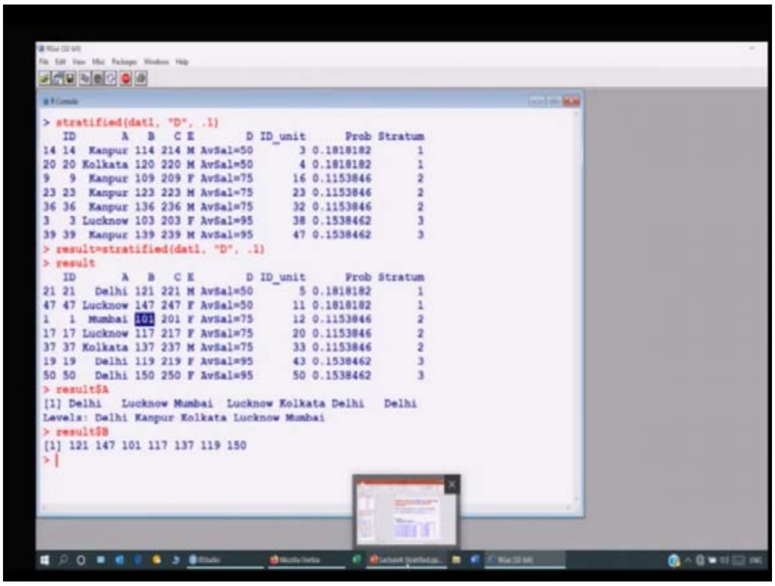
(Refer Slide Time: 30:45)



```
43 43 Lucknow 243 AvSal=50 F 143       43    1       43
44 44 Lucknow 244 AvSal=75 F 144       44    1       44
45 45  Kanpur 245 AvSal=95 M 145       45    1       45
46 46   Delhi 246 AvSal=50 M 146       46    1       46
47 47 Lucknow 247 AvSal=50 F 147       47    1       47
48 48   Delhi 248 AvSal=75 M 148       48    1       48
49 49  Mumbai 249 AvSal=75 M 149       49    1       49
50 50   Delhi 250 AvSal=95 F 150       50    1       50
> stratified(dat1, "B", .1)
   ID       A   C        D E   B ID_unit Prob Stratum
1   1  Mumbai 201 AvSal=75 F 101       1    1       1
2   2  Kanpur 202 AvSal=75 F 102       2    1       2
3   3 Lucknow 203 AvSal=95 F 103       3    1       3
4   4 Lucknow 204 AvSal=95 M 104       4    1       4
5   5  Mumbai 205 AvSal=75 M 105       5    1       5
6   6  Kanpur 206 AvSal=50 M 106       6    1       6
7   7   Delhi 207 AvSal=75 M 107       7    1       7
8   8   Delhi 208 AvSal=95 F 108       8    1       8
9   9  Kanpur 209 AvSal=75 F 109       9    1       9
10 10 Kolkata 210 AvSal=75 M 110      10    1      10
11 11  Kanpur 211 AvSal=75 F 111      11    1      11
12 12  Mumbai 212 AvSal=50 F 112      12    1      12
13 13 Lucknow 213 AvSal=95 M 113      13    1      13
14 14  Kanpur 214 AvSal=50 M 114      14    1      14
15 15  Mumbai 215 AvSal=75 M 115      15    1      15
16 16  Kanpur 216 AvSal=95 F 116      16    1      16
17 17 Lucknow 217 AvSal=75 F 117      17    1      17
```

And if I try to make it here just 10 percent, you can see here this is my 10 percent sample, right. Because, why the case is happening? Because you can see here means everything is coming here 50 because B has been taken as the values 1 to 50. So, I mean there is no variability, right, all the values 1 to 50 that will create 50 strata. So, this is exactly what I wanted to show you, but instead if you try to obtain this sample with respect to D, then it will try to stratify it.

(Refer Slide Time: 31:18)



```
> stratified(dat1, "D", .1)
   ID       A   B   C E        D ID_unit      Prob Stratum
14 14  Kanpur 114 214 M AvSal=50       3 0.1818182       1
20 20 Kolkata 120 220 M AvSal=50       4 0.1818182       1
9   9  Kanpur 109 209 F AvSal=75      16 0.1153846       2
23 23  Kanpur 123 223 M AvSal=75      23 0.1153846       2
36 36  Kanpur 136 236 M AvSal=75      32 0.1153846       2
3   3 Lucknow 103 203 F AvSal=95      38 0.1538462       3
39 39  Kanpur 139 239 M AvSal=95      47 0.1538462       3
> result=stratified(dat1, "D", .1)
> result
   ID       A   B   C E        D ID_unit      Prob Stratum
21 21   Delhi 121 221 M AvSal=50       5 0.1818182       1
47 47 Lucknow 147 247 F AvSal=50      11 0.1818182       1
1   1  Mumbai 101 201 F AvSal=75      12 0.1153846       2
17 17 Lucknow 117 217 F AvSal=75      20 0.1153846       2
37 37 Kolkata 137 237 M AvSal=75      33 0.1153846       2
19 19   Delhi 119 219 F AvSal=95      43 0.1538462       3
50 50   Delhi 150 250 F AvSal=95      50 0.1538462       3
> result$A
[1] Delhi   Lucknow Mumbai  Lucknow Kolkata Delhi   Delhi
Levels: Delhi Kanpur Kolkata Lucknow Mumbai
> result$B
[1] 121 147 101 117 137 119 150
> |
```

So, you can see here if I clear this screen and instead of here D, we I try to give it here D, then you can see here this sample is here, right ok. And after that if you try to save this result, as a result is equal to or I can say here whatever is my outcome, this is saved here as a result equal to like this. So, you can see here now this is the data frame, which I have fixed.

And now if I want to get the information on particular variable say here, on say I want to extract the information on the variable a from the sample. So, result A you can see here this is giving me the same cities which are mentioned over here. And similarly, if I want to have the details on the variable B from this, you can see here this is 121, 147, 101 etc.

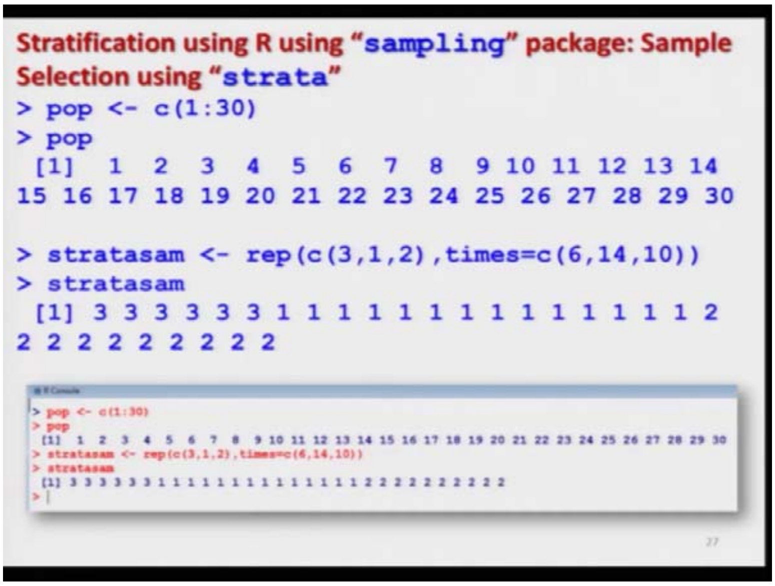
And this is the same thing which is given here under B, 121, 147, 101 etc. So, you can see here, this is how you can extract the information on a particular variable, also from this data set, right. So, now let me come back to our slide and we try to go for another package.

So, once again we will use the same package sampling, but there is a different command, using strata to draw the stratified sample, right. So, here I have created a population. So, you can see here the population values are given in this column and they are simply the numbers from 1, 2, 3, 4 up to here 30.

And then I have intentionally divided it, actually that will be the need what we have to enter in the program in a strata, that I have to first specify that how I am going to allocate my units to different stratum.

So, what I have done? I have created three srtata here and what I am saying here, just to create more heterogeneity that the variables the values 1 to 6 they will be under stratum number 3, the values from 7 to here 20, they will be under stratum number 1 and the remaining values from 21 to 30 they will be under the stratum number 2. And I have just given it a different color, so that you can clearly see it, right.

(Refer Slide Time: 34:08)



Stratification using R using "sampling" package: Sample Selection using "strata"

```
> pop <- c(1:30)
> pop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

> stratasam <- rep(c(3,1,2),times=c(6,14,10))
> stratasam
 [1] 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
2 2 2 2 2 2 2 2 2
```

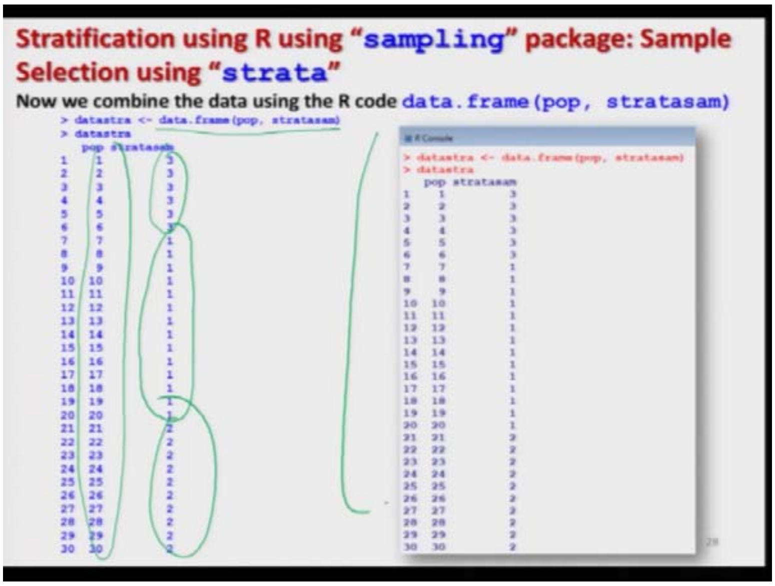
```
> pop <- c(1:30)
> pop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
> stratasam <- rep(c(3,1,2),times=c(6,14,10))
> stratasam
 [1] 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
> |
```

27

And then I would try to draw the stratified sample from this population. So, in order to create this data set let me create here the variable pop which is the number from 1 to 30 you can see here. And then I try to include the values here which are given in the stratum number. So, 1, 2, 3, 4, 5, 6 three are 6 values.

Similarly, there are 1 R number in 14 and 2 is repeated 10 times. So, I create another variable here, state stratasam. And this comes out to be like this and then I try to club them together in the framework of a data frame.

(Refer Slide Time: 34:45)



Stratification using R using "sampling" package: Sample Selection using "strata"

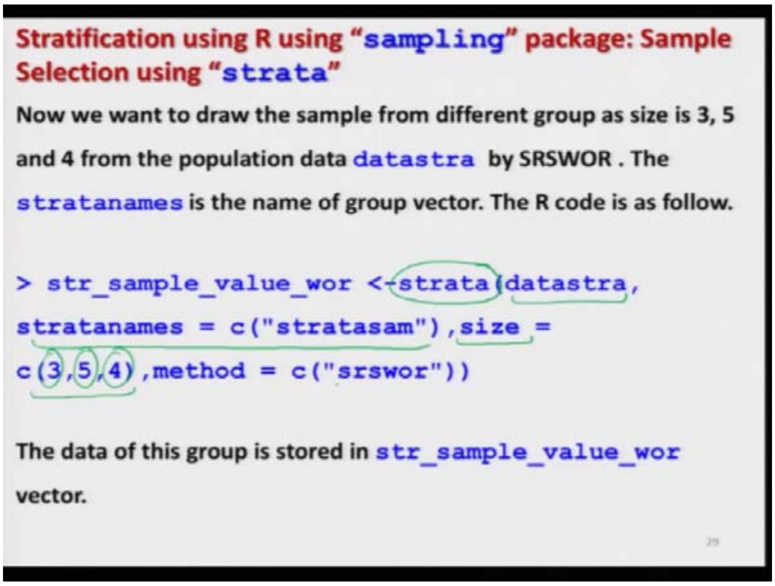
Now we combine the data using the R code `data.frame(pop, stratasam)`

```
> datastra <- data.frame(pop, stratasam)
> datastra
   pop stratasam
1    1         3
2    2         3
3    3         3
4    4         3
5    5         3
6    6         3
7    7         1
8    8         1
9    9         1
10  10         1
11  11         1
12  12         1
13  13         1
14  14         1
15  15         1
16  16         1
17  17         1
18  18         1
19  19         1
20  20         1
21  21         2
22  22         2
23  23         2
24  24         2
25  25         2
26  26         2
27  27         2
28  28         2
29  29         2
30  30         2
```

```
> datastra <- data.frame(pop, stratasam)
> datastra
   pop stratasam
1    1         3
2    2         3
3    3         3
4    4         3
5    5         3
6    6         3
7    7         1
8    8         1
9    9         1
10  10         1
11  11         1
12  12         1
13  13         1
14  14         1
15  15         1
16  16         1
17  17         1
18  18         1
19  19         1
20  20         1
21  21         2
22  22         2
23  23         2
24  24         2
25  25         2
26  26         2
27  27         2
28  28         2
29  29         2
30  30         2
```

So, I am simply creating this data which is given in the table here like this. So, you can see here now the I have here sample values from 1 to 30. And then they have been allocated to stratum number 3, then 1 and then here 2 and this is here the screenshot.

(Refer Slide Time: 35:05)



Stratification using R using "sampling" package: Sample Selection using "strata"

Now we want to draw the sample from different group as size is 3, 5 and 4 from the population data `datastra` by SRSWOR . The `stratanames` is the name of group vector. The R code is as follow.

```
> str_sample_value_wor <-strata(datastra,
stratanames = c("stratasam"),size =
c(3,5,4),method = c("srswor"))
```

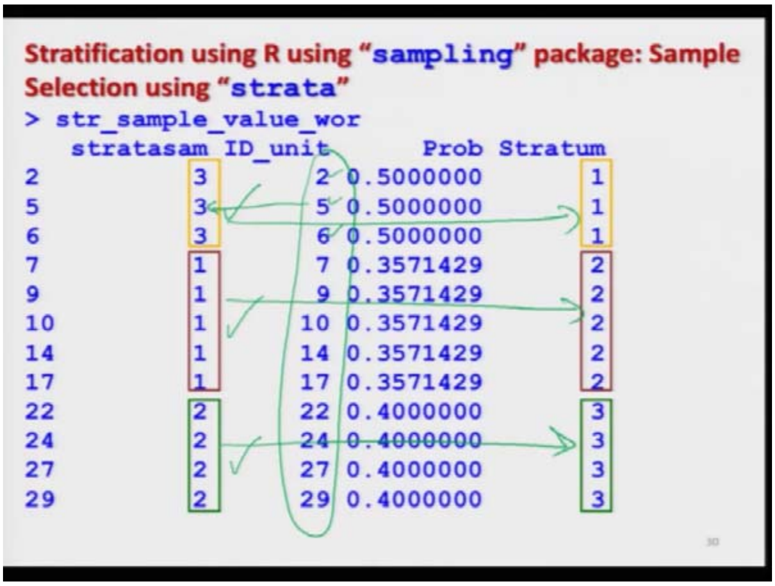
The data of this group is stored in `str_sample_value_wor` vector.

Now, from this population, we will try to draw the sample. And for that we will use the command here C strata ok and this data population data has been stored in a variable, say data s t r a. So, now I will say here, I will take here some example to show you, how the things are going to be there. So, first you try to use the command here strata.

Then you have to give the name of the data frame, data s t r a and then you have to give the name of the variable with which you want to stratify and then you have to give the size of the samples which you want to draw. Suppose I have given here 3, 5, 4 that means, I want to draw sample of size 3 from the stratum number 1.

What was my first strata? And then sample of size 5 from the 2nd strata and sample of size 4 from the 3rd is strata and this I want to do with respect to simple random sampling without replacement, right. And whatever are the values which I am that I am trying to store in this variable name s t r underscore sample, underscore value w o r, that means stratified sample values under w o r.

(Refer Slide Time: 36:24)



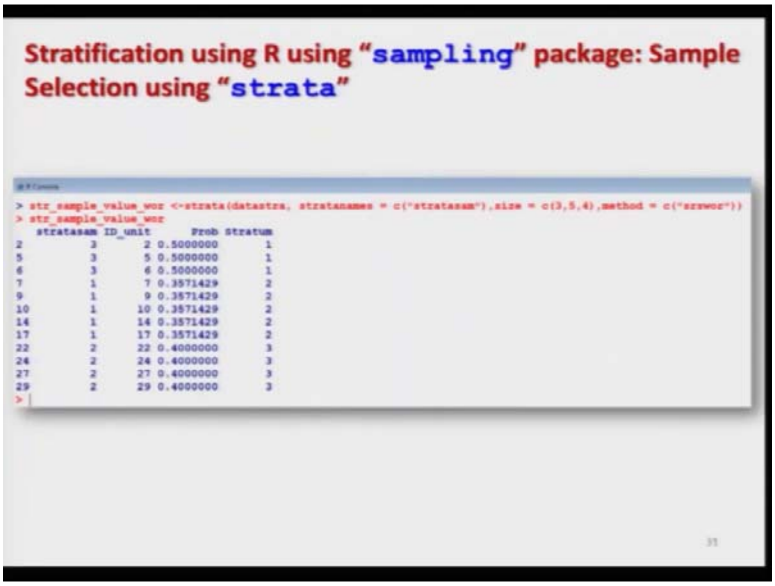
Stratification using R using "sampling" package: Sample Selection using "strata"

```
> str_sample_value_wor
   stratasam ID_unit      Prob Stratum
2          3       2 0.5000000       1
5          3       5 0.5000000       1
6          3       6 0.5000000       1
7          1       7 0.3571429       2
9          1       9 0.3571429       2
10         1      10 0.3571429       2
14         1      14 0.3571429       2
17         1      17 0.3571429       2
22         2      22 0.4000000       3
24         2      24 0.4000000       3
27         2      27 0.4000000       3
29         2      29 0.4000000       3
```

And when I try to execute it on the R console, I get here an outcome like this one. So, first you can see here the screenshot, a screenshot will look like here this, which I have modified a little bit here to explain you. So, you can see here I am trying to take here a sample of psi 3, 5 and 4. So, you can see here I have got here sample of size 3, sample of size 5 and sample of size 4. And this is coming from stratum number 1, this is coming from stratum number 2 and this is coming from stratum number 3.

And what are the units which I have obtained here? The units which are obtained are here given in this column. So, unit number 2, 5 and 7, they are coming from stratum number 3. So, you can see here in this table itself, that strata unit number 2, 3, 4 etc., they are in stratum number 3. So, this is how we try to get here a sample.

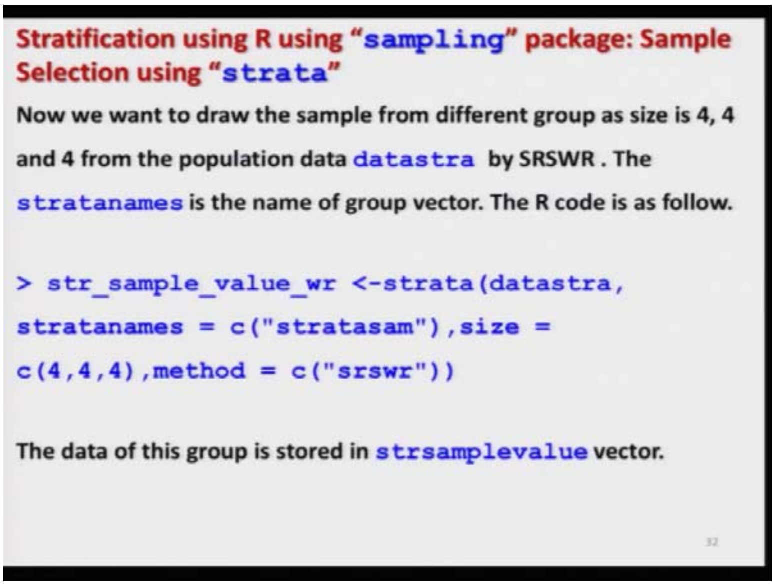
(Refer Slide Time: 37:31)



Stratification using R using "sampling" package: Sample Selection using "strata"

```
> str_sample_value_wor <-strata(datastra, stratanames = c("stratasam"),size = c(3,5,4),method = c("srswor"))
> str_sample_value_wor
   stratasam ID_unit      Prob Stratum
2          3       2 0.5000000       1
5          3       5 0.5000000       1
6          3       6 0.5000000       1
7          1       7 0.3571429       2
9          1       9 0.3571429       2
10         1      10 0.3571429       2
14         1      14 0.3571429       2
17         1      17 0.3571429       2
22         2      22 0.4000000       3
24         2      24 0.4000000       3
27         2      27 0.4000000       3
29         2      29 0.4000000       3
>
```

(Refer Slide Time: 37:33)



Stratification using R using "sampling" package: Sample Selection using "strata"

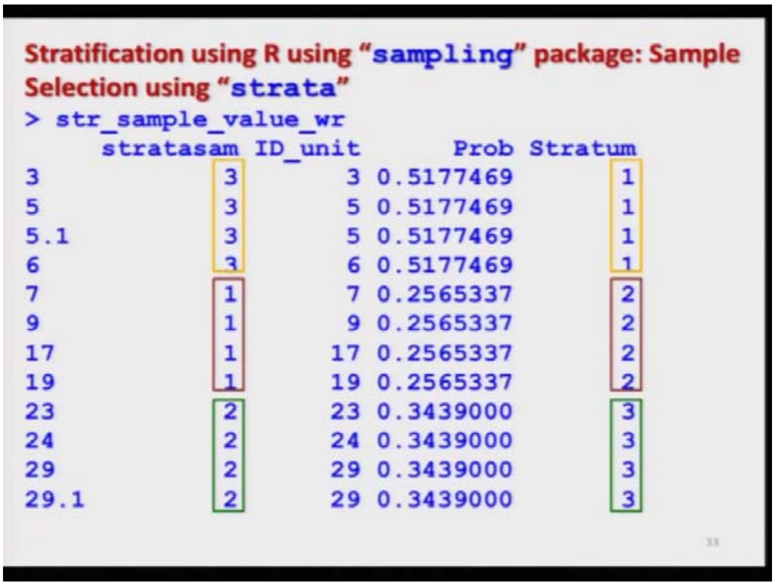
Now we want to draw the sample from different group as size is 4, 4 and 4 from the population data datastra by SRSWR . The stratanames is the name of group vector. The R code is as follow.

```
> str_sample_value_wr <-strata(datastra,
stratanames = c("stratasam"),size =
c(4,4,4),method = c("srswr"))
```

The data of this group is stored in strsamplevalue vector.

Now, similarly if you want to use here simple random sampling with the replacement, so you try to use the same command, means everything remains the here same. And suppose we want to draw the samples of size 4 each from each of the stratum 1, 2 and 3. So, this is what we have to change here and then we have to change the name here srswr, because now we want to have sample from with the replacement.

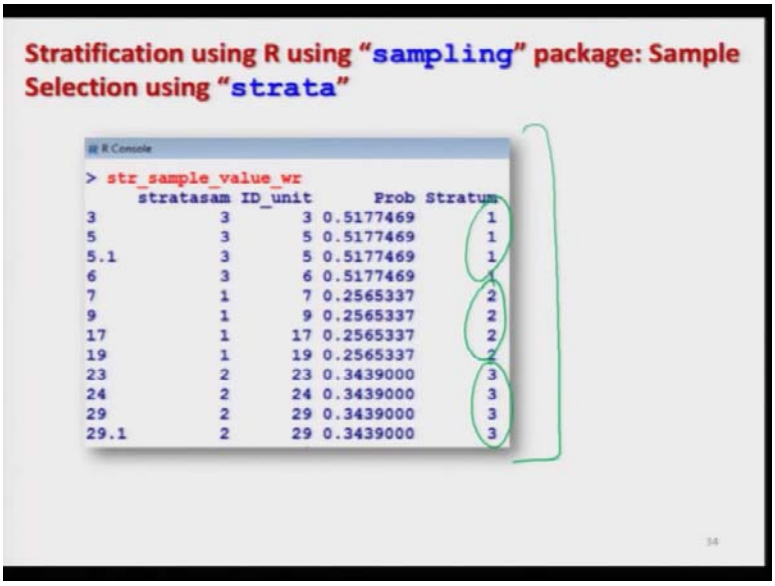
(Refer Slide Time: 38:00)



Stratification using R using "sampling" package: Sample Selection using "strata"

```
> str_sample_value_wr
     stratasam ID_unit      Prob Stratum
3            3       3 0.5177469       1
5            3       5 0.5177469       1
5.1          3       5 0.5177469       1
6            3       6 0.5177469       1
7            1       7 0.2565337       2
9            1       9 0.2565337       2
17           1      17 0.2565337       2
19           1      19 0.2565337       2
23           2      23 0.3439000       3
24           2      24 0.3439000       3
29           2      29 0.3439000       3
29.1         2      29 0.3439000       3
```

(Refer Slide Time: 38:06)



Stratification using R using "sampling" package: Sample Selection using "strata"

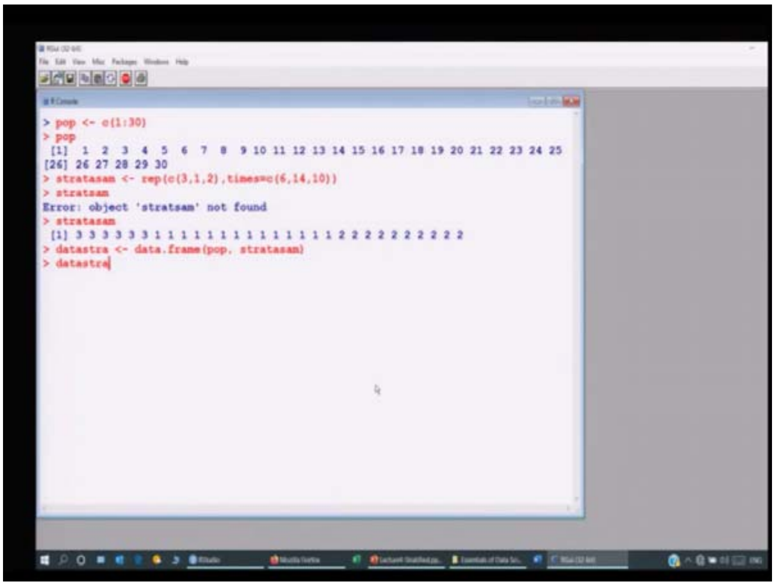
```
> str_sample_value_wr
     stratasam ID_unit      Prob Stratum
3            3       3 0.5177469       1
5            3       5 0.5177469       1
5.1          3       5 0.5177469       1
6            3       6 0.5177469       1
7            1       7 0.2565337       2
9            1       9 0.2565337       2
17           1      17 0.2565337       2
19           1      19 0.2565337       2
23           2      23 0.3439000       3
24           2      24 0.3439000       3
29           2      29 0.3439000       3
29.1         2      29 0.3439000       3
```

And if you try to do it here, you can see here this is the outcome. This outcome will actually look like this, this is the screenshot and you can see here, there are 4 values, 4 values and 4 values here, which I have explained here.

So, you can see here there are 4 values from the stratum number ID 3, 4 values from 1, 4 values from 2 and so and this is indicating that 4 values are coming from stratum 1st stratum, 2nd stratum and 3rd stratum and these are the values, which are selected in your sample.

So, from this if you want to recover the values of this data, you can simply use the same command that name of the data frame, then the name of the variable and join by the dollar sign, right. So, you can see here. Now, let me come back to the R console and try to show you how the things are going to work, right.

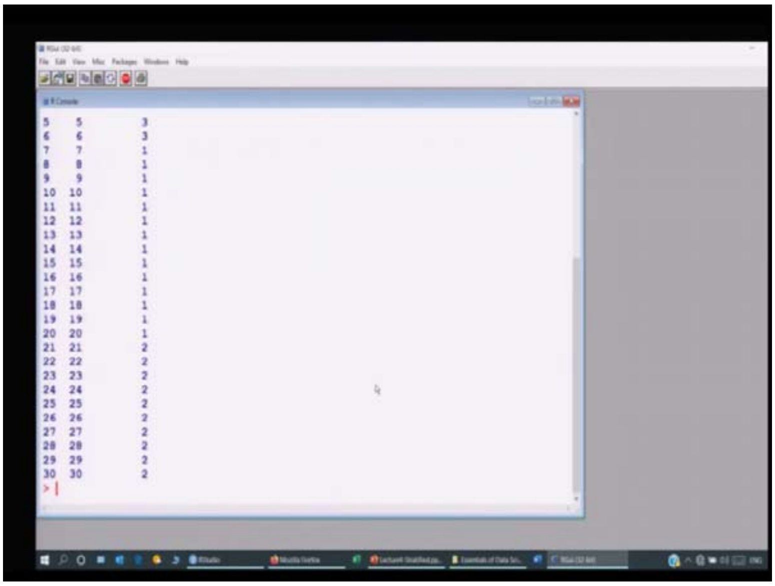
(Refer Slide Time: 39:03)



```
> pop <- c(1:30)
> pop
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30
> stratasam <- rep(c(3,1,2),times=c(6,14,10))
> stratsam
Error: object 'stratsam' not found
> stratasam
 [1] 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
> datastra <- data.frame(pop, stratasam)
> datastra
```

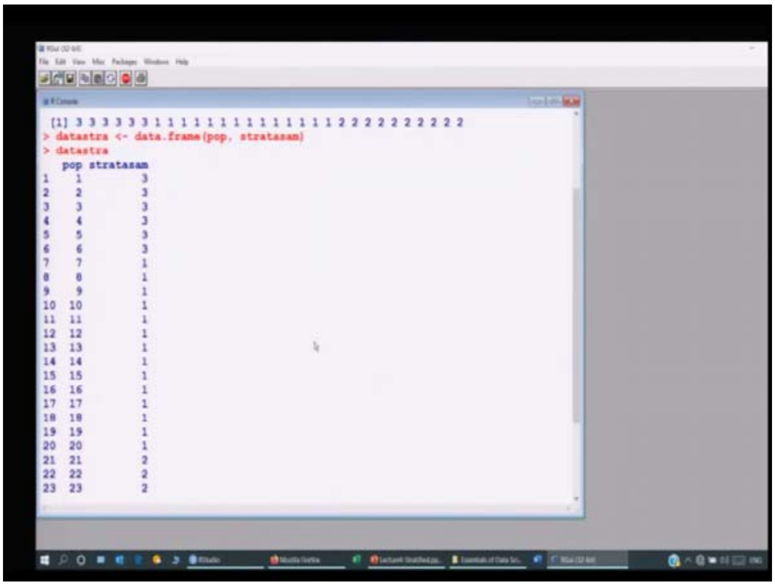
So, first I have entered here the population value, you can see here this is your here pop then this is another variable for the stratum number, you can see here it is sorry stratasam this is here. And then I try to create here a data frame of these two things. So, you can see here this is my here data frame from where I would like to draw the sample.

(Refer Slide Time: 39:31)



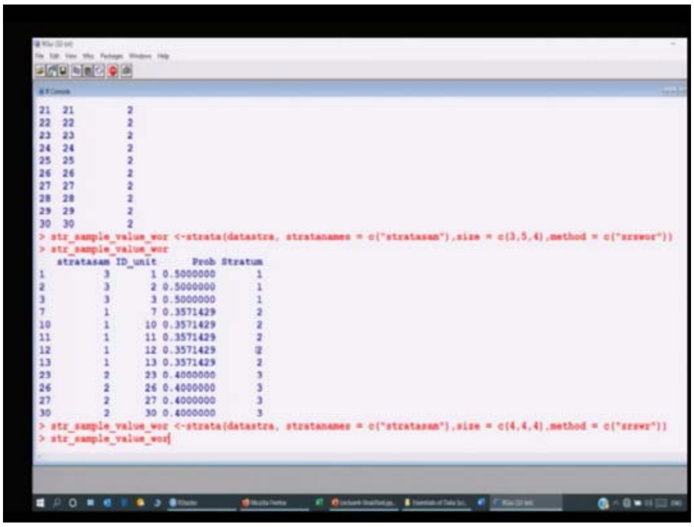
```
5    5         3
6    6         3
7    7         1
8    8         1
9    9         1
10  10         1
11  11         1
12  12         1
13  13         1
14  14         1
15  15         1
16  16         1
17  17         1
18  18         1
19  19         1
20  20         1
21  21         2
22  22         2
23  23         2
24  24         2
25  25         2
26  26         2
27  27         2
28  28         2
29  29         2
30  30         2
> |
```

(Refer Slide Time: 39:33)



```
 [1] 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
> datastra <- data.frame(pop, stratasam)
> datastra
   pop stratasam
1    1         3
2    2         3
3    3         3
4    4         3
5    5         3
6    6         3
7    7         1
8    8         1
9    9         1
10  10         1
11  11         1
12  12         1
13  13         1
14  14         1
15  15         1
16  16         1
17  17         1
18  18         1
19  19         1
20  20         1
21  21         2
22  22         2
23  23         2
```

(Refer Slide Time: 39:44)

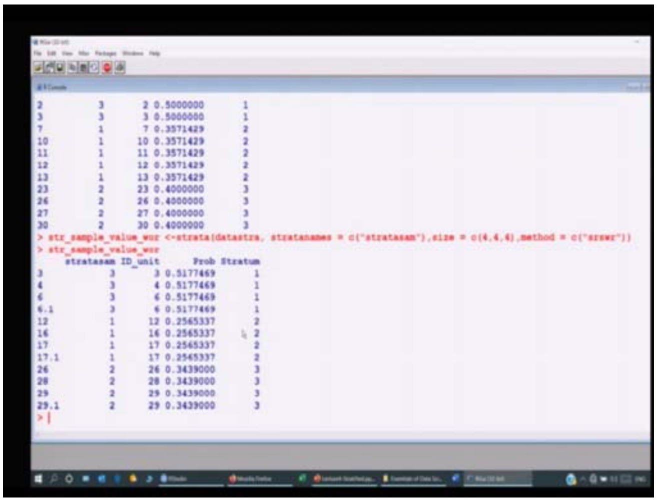


```
21  21          2
22  22          2
23  23          2
24  24          2
25  25          2
26  26          2
27  27          2
28  28          2
29  29          2
30  30          2
> str_sample_value_wor <-strata(datastra, stratanames = c("stratasam"),size = c(3,5,4),method = c("srswor"))
> str_sample_value_wor
   stratasam ID_unit      Prob Stratum
1          3       1 0.5000000       1
2          3       2 0.5000000       1
3          3       3 0.5000000       1
7          1       7 0.3571429       2
10         1      10 0.3571429       2
11         1      11 0.3571429       2
12         1      12 0.3571429       2
13         1      13 0.3571429       2
23         2      23 0.4000000       3
26         2      26 0.4000000       3
27         2      27 0.4000000       3
30         2      30 0.4000000       3
> str_sample_value_wor <-strata(datastra, stratanames = c("stratasam"),size = c(4,4,4),method = c("srswr"))
> str_sample_value_wor
```

Now, I try to use the command here to draw the sample and if I try to paste it here, you can see here if I try to see, what is the outcome. So, I try to see, this is the sample which I have got here, right. So, you can see here this is the sample of size 3, 5, 4 from the stratum number 3, 1 and 2 respectively, which are corresponding to 1st strata, 2nd strata and 3rd strata in the last column.

And similarly. if I want to have here a sample of size 4 from each of the strata, then I have to and by srswr. So, I just modify my sample sizes and change method is equal to srswr within the double quotes, and if you try to see here, now the outcome here is like this, right.

(Refer Slide Time: 40:42)



```
2          3       2 0.5000000       1
3          3       3 0.5000000       1
7          1       7 0.3571429       2
10         1      10 0.3571429       2
11         1      11 0.3571429       2
12         1      12 0.3571429       2
13         1      13 0.3571429       2
23         2      23 0.4000000       3
26         2      26 0.4000000       3
27         2      27 0.4000000       3
30         2      30 0.4000000       3
> str_sample_value_wor <-strata(datastra, stratanames = c("stratasam"),size = c(4,4,4),method = c("srswr"))
> str_sample_value_wor
     stratasam ID_unit      Prob Stratum
3            3       3 0.5177469       1
4            3       4 0.5177469       1
6            3       6 0.5177469       1
6.1          3       6 0.5177469       1
12           1      12 0.2565337       2
16           1      16 0.2565337       2
17           1      17 0.2565337       2
17.1         1      17 0.2565337       2
26           2      26 0.3439000       3
28           2      28 0.3439000       3
29           2      29 0.3439000       3
29.1         2      29 0.3439000       3
> |
```

So, this is a sample from the srswr. So, now I have given you the demonstration that how you can draw the stratified random sample using these two different types of commands strata and a stratified.R. Now they have their own advantages and their own limitations. Once now you have obtained the sample.

Then you have to write a very simple program to compute the mean or the estimate of the variances, they are extremely simple, right. As far as I know, I have not found these direct commands in these two functions, but there is a 3rd package in which that draws the sample and also gives you the estimate of population mean and standard errors.

So, that I will try to discuss in the next lecture, but that will also have its own limitations and different commands, different function, different ways of presentation. So, you take some example try to create a population yourself and try to draw the sample. And try to match, that whatever you have learnt in the theory is that really happening, right.

I have not given you here command, how to draw the sample using the optimum allocation, because they are not really available, up to now in any this function because they require the value of capital S_i, right.

So, means if you want it and if you know the value of S_i, you can write a simple program that is not difficult. Because you have to just keep in mind that from every strata you are trying to draw the simple random sample. So, it is not difficult to write a small function yourself also, if you want to really implement it. So, you try to practice take some example and I will see you in the next lecture, till then good bye.