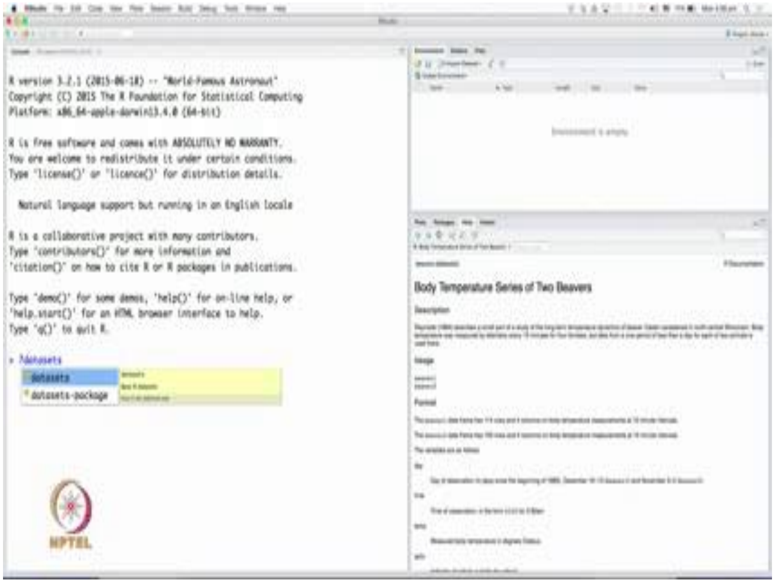
Introduction to Research
Prof. Arun K. Tangirala
Department of Metallurgical and Materials Engineering
Indian Institute of Technology, Madras

Lecture – 15 (2 B)
Data Analysis

Welcome to the demonstration part of the Data Analysis lecture, where I am going to show you how to just analyze a couple of data sets in a package or software package called R, and as I had said R is an open source and free software, very popular. It's available on all platforms. I am not going to show you how to install R. You can just go to the website, download R; and for R, there is a very nice graphical user interface - GUI - called the R studio, and I strongly recommend you download that and install. All of that should be done fairly easily. I have done that on my computer. So, if you have a computer handy, you may want pause this video, install, and then, come back, and then, carry on with the instructions that I go through in this lecture.

(Refer Slide Time: 01:16)

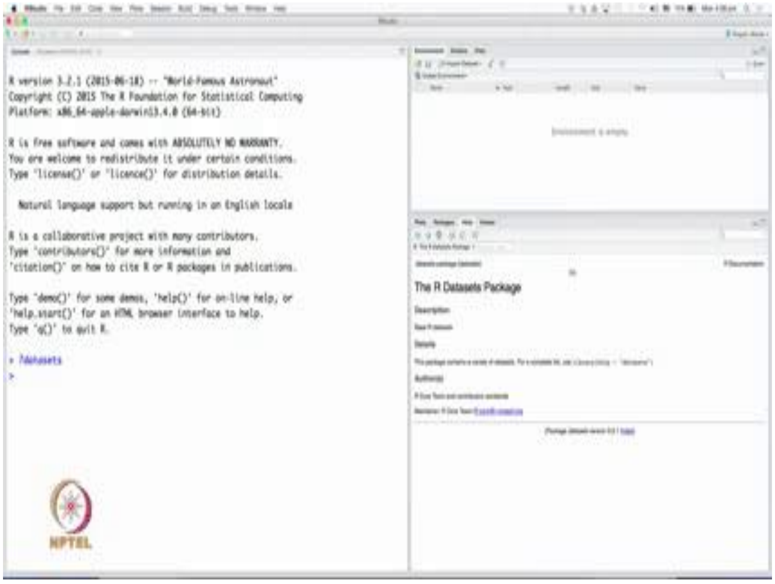


```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
```

So, I have R studio installed. Of course, before installing R studio, I have installed R as well and I am going to open this R studio interface. This is how the interface should look like. Of course, it can look a bit different depending on how you configure it. The purpose of this lecture is not to show you what R is about and so on. There are many, a number of, tutorials available on the web for that purpose. The purpose is to just show you what careful data analysis is about, just with a couple of data sets. Now, as I had mentioned during my lecture, the R package comes with a lot of data sets that the user can clear on with. That's one of the nice features of R.

Now, to know what data sets are available, one could use this syntax in R. The question marks syntax in R basically brings up help, and the nice thing about R studio is, it completes the keyword or whatever you are typing if it is a part of the R system and data sets is a package in R. Therefore, as I typed data sets, you can see it's trying to complete that for me and its also saying that is a package. So, what I am typing, therefore, is right; what I am looking for, it makes sense.

(Refer Slide Time: 02:34)



```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
>
```

Environment is empty

datasets-package {datasets}

R Documentation

The R Datasets Package

Description

Base R datasets

Details

This package contains a variety of datasets. For a complete list, use library(help = "datasets")

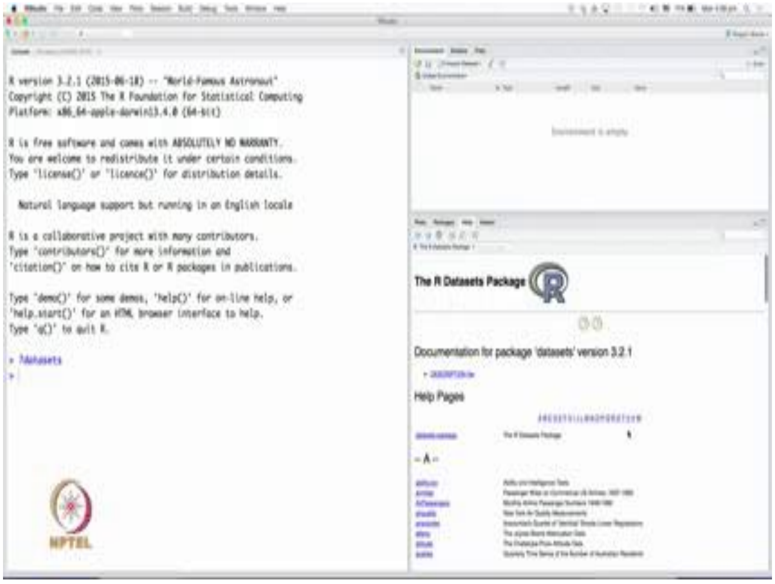
Author(s)

R Core Team and contributors worldwide

Maintainer: R Core Team R-core@r-project.org

[Package datasets version 3.2.1 Index]

(Refer Slide Time: 02:46)



```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
>
```

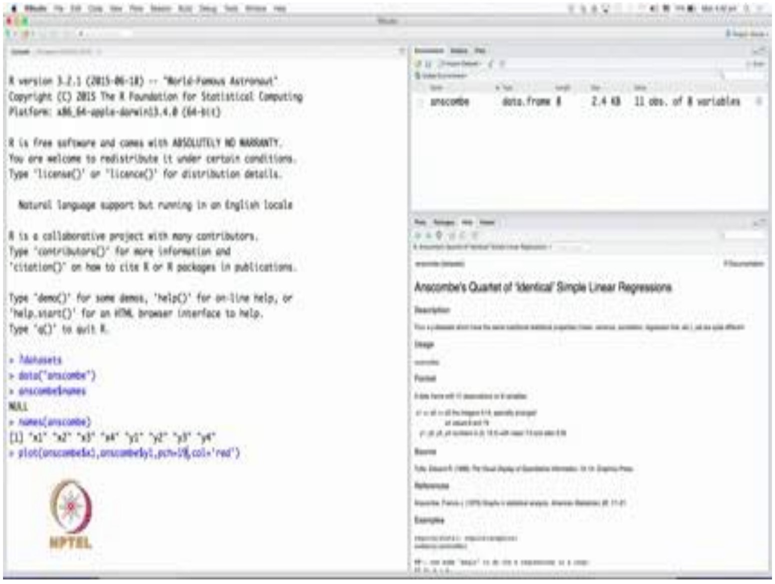
The R Datasets Package

Documentation for package 'datasets' version 3.2.1

Help Pages

Let's ask for help on data sets and it says, R data sets is a package. To know what data sets are available in R, go to the index at the bottom here, and that brings up the documentation for the package data set. I have the version 3.2.1 and they are listed in alphabetical order. Of course, it's impossible for me to go over all these data sets.

(Refer Slide Time: 03:13)



```
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
> data("anscombe")
> anscombe$names
NULL
> names(anscombe)
[1] "x1" "x2" "x3" "x4" "y1" "y2" "y3" "y4"
> plot(anscombe$x1,anscombe$y1,pch=19,col='red')
```

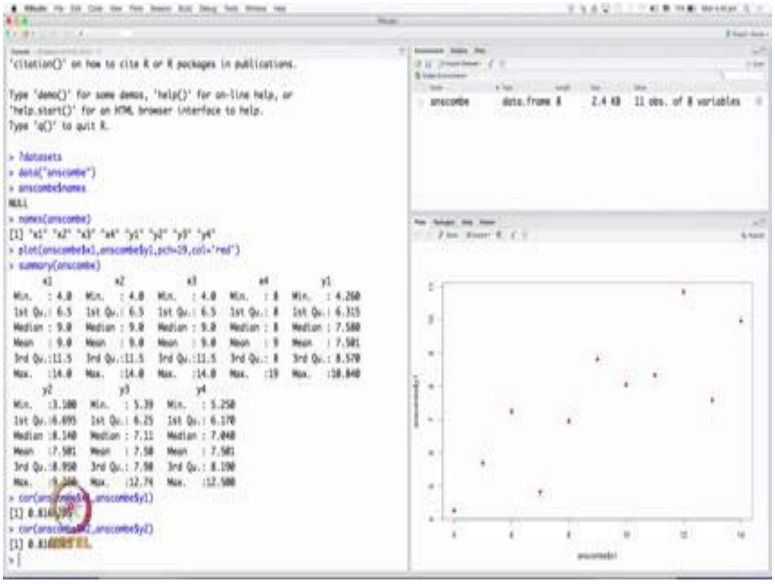
```
anscombe    data.frame 8    2.4 KB    11 obs. of 8 variables
```

Anscombe's Quartet of 'Identical' Simple Linear Regressions

Let's look at this particular data set called Beavers, but before I do that, I want to draw your attention to this Anscombe data set that we talked about in the lecture. If you click on the Anscombe data set, it gives you a description of the data set. In fact, we can load that to begin with. To load a data set that comes with the R package, all you have to do is, type data, and then type the name of the data set, and once again R studio shows the list of the completions that one can have, and Anscombe is the one that I am looking for, and I choose that. Once I do that, as you can see on the top right, Anscombe data set has been loaded. Don't worry about the promise part; it's just not exactly loaded. The moment you talk about Anscombe or in the sense in the command prompt, the moment you start typing the name of the data set, it shows you the true nature of the data - whether it is a time series data or regular data and so on. It shows that is a data frame. A data frame in R is nothing but a matrix of data, but with the columns labeled, and the labels of Anscombe or any other data frame can be easily found.

For example, we could ask what are the labels for Anscombe data set. This is the syntax. So, these are the labels that go with the columns. Anscombe data set has 8 columns. Remember we had four pairs 4 x(s) and 4 y(s) and one could plot the x 1 verses y 1 or x 2 y 2 and regenerate the plots that I showed you in the lecture. For example, if I would, if I want to plot the Anscombe x 1 y 1 pair, then I could, I am just recalling the commands that I have in my history and that's a nice feature of R studio. So, here I am plotting x 1, sorry y 1 verses x 1 of the Anscombe data set, and the syntax here tells me PCH equals 19; these are all optional syntaxes. If I omit those, then it will just simply produce a scatter plot, but I don't, I want something more. I want solid circles that are colored red.

(Refer Slide Time: 05:36)



```
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
> data("anscombe")
> anscombe$names
NULL
> names(anscombe)
[1] "x1" "x2" "x3" "x4" "y1" "y2" "y3" "y4"
> plot(anscombe$x1,anscombe$y1,pch=19,col='red')
> summary(anscombe)
       x1             x2             x3             x4           y1        
 Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8   Min.   : 4.260  
 1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8   1st Qu.: 6.315  
 Median : 9.0   Median : 9.0   Median : 9.0   Median : 8   Median : 7.580  
 Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9   Mean   : 7.501  
 3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8   3rd Qu.: 8.570  
 Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19   Max.   :10.840  
       y2              y3              y4        
 Min.   :3.100   Min.   : 5.39   Min.   : 5.250  
 1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170  
 Median :8.140   Median : 7.11   Median : 7.040  
 Mean   :7.501   Mean   : 7.50   Mean   : 7.501  
 3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190  
 Max.   :9.260   Max.   :12.74   Max.   :12.500  
> cor(anscombe$x1,anscombe$y1)
[1] 0.8164205
> cor(anscombe$x2,anscombe$y2)
[1] 0.8162365
> |
```

```
anscombe      data.frame 8      2.4 KB    11 obs. of 8 variables
```

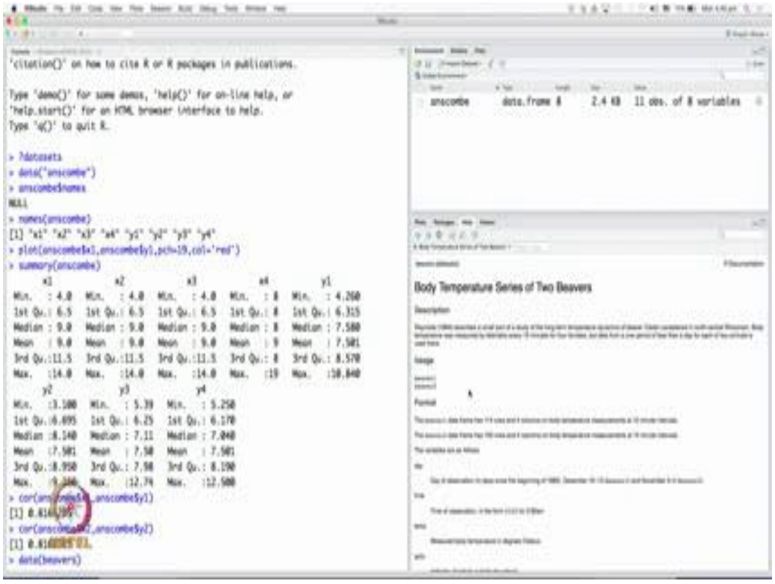
So, PCH19 is basically telling which character it should use for plotting the markers, and that, and the color tells the plot to use a red color. So, if I do this, then it produces as you see on the right bottom screen, I have here the y 1 verses x 1. As you can see on the y and x axis labels, you can also, of course, enhance and decorate these plots by providing x y labels. I am not going to go into that, but main thing that you should observe is this plot is exactly identical to what you have seen in the lecture and I welcome you to reproduce the other plots that I have shown you in the lecture.

And of course, what one could ask for is the summary statistics. Summary statistics are nothing but mean, median, and so on. So, we could ask for the summary statistics for the Anscombe data set. The nice thing about the summary command in R is that for a data frame, it looks through every column and reports the mean, minimum, maximum, median and so on. Particular attention that I want to draw your attention to is the mean. So, mean of x 1, x 2, y 2 or if you can look at mean of y 1 and mean of y 2 and y 3, y 4 are identical; may be mean of y 3 is just differing by one-third decimal. They all have the same average.

In fact, you can also check if they have the same correlation; that is x 1 and y 1 have the same correlation, x 2 and y 2 have the same correlation and so on. So, how do you compute correlation? Let’s say that we want to compute the correlation between x 1 and y 1 of the Anscombe data set. This dollar operator is an operator in R which allows you to access the columns of a data frame. Data frame is different from matrix but it looks like a matrix, but the dollar operator applies to data frames, and lists, and so on. So, this is the correlation between x 1 and y 1. Let’s see if they have the same - x 2 and y 2 have the same correlation; so exactly the same correlation. So, every pair in the Anscombe data set has the same correlation despite looking strikingly different.

Now, let’s move on to the next data set that I want to show you. In fact, the nice thing about R is that it comes, although it comes with a few installed packages by default, one can go to the R website and look for packages of interest from the thousands of user contributed packages meant for additional purposes and there are many, many such packages, play around with them and the other nice thing is each package comes with a data set that you can test the routines in that package. Again, if you go to the R tutorial, you can realize how to do that.

(Refer Slide Time: 08:57)



```
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
> data("anscombe")
> anscombe$names
NULL
> names(anscombe)
[1] "x1" "x2" "x3" "x4" "y1" "y2" "y3" "y4"
> plot(anscombe$x1,anscombe$y1,pch=19,col='red')
> summary(anscombe)
       x1             x2             x3             x4           y1
 Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8   Min.   : 4.260
 1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8   1st Qu.: 6.315
 Median : 9.0   Median : 9.0   Median : 9.0   Median : 8   Median : 7.580
 Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9   Mean   : 7.501
 3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8   3rd Qu.: 8.570
 Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19   Max.   :10.840
       y2              y3              y4
 Min.   :3.100   Min.   : 5.39   Min.   : 5.250
 1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170
 Median :8.140   Median : 7.11   Median : 7.040
 Mean   :7.501   Mean   : 7.50   Mean   : 7.501
 3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190
 Max.   :9.260   Max.   :12.74   Max.   :12.500
> cor(anscombe$x1,anscombe$y1)
[1] 0.816[illegible]
> cor(anscombe$x2,anscombe$y2)
[1] 0.816[illegible]
> data(beavers)
```

```
anscombe     data.frame 8     2.4 KB   11 obs. of 8 variables
```

Body Temperature Series of Two Beavers

Description

Usage

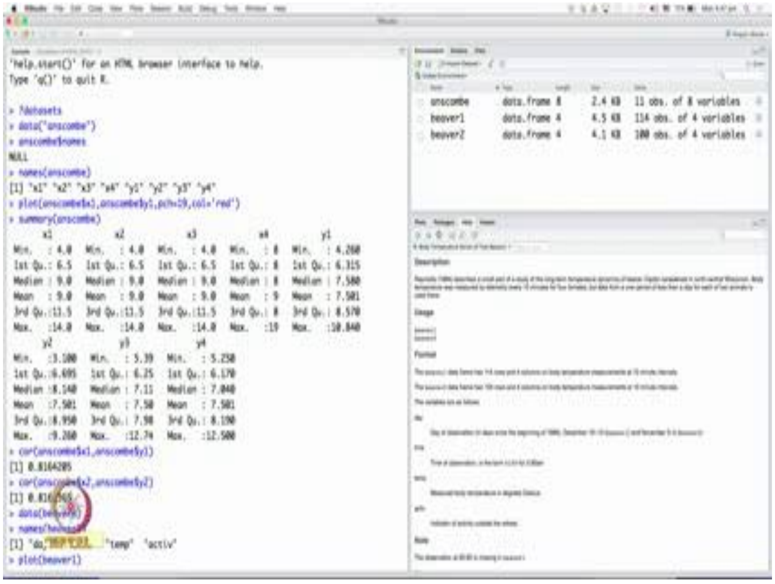
Format

So, let’s load another package called Beavers, and to know what these Beavers data set is about, we will go back to the help, and pick the Beaver’s package. It says it’s a body temperature series of two beavers. Beavers are this huge rodents that are found in North

America and so on, and they are very well known for building dams, small dams, and walls, and so on. So, they are kind of a beaver in that sense.

So, these Beavers has two data sets in it, beaver1 and beaver2 corresponding to the body temperature of two different beavers, and we look at the first data set - the beaver1; it has about 114 data points in it, collected at 10 minute intervals using a technique called telemetry and there is more information on this in the documentation.

(Refer Slide Time: 09:43)

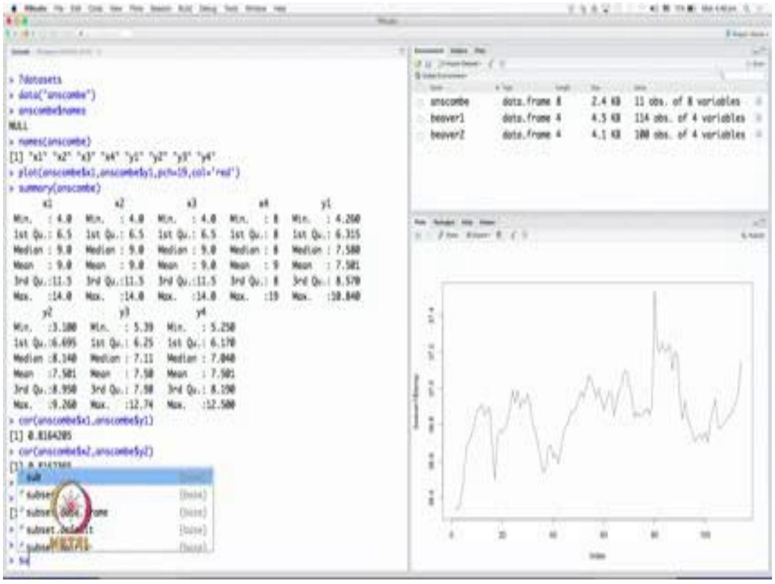


```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?datasets
> data("anscombe")
> anscombe$names
NULL
> names(anscombe)
[1] "x1" "x2" "x3" "x4" "y1" "y2" "y3" "y4"
> plot(anscombe$x1,anscombe$y1,pch=19,col='red')
> summary(anscombe)
       x1             x2             x3             x4           y1
 Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8   Min.   : 4.260
 1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8   1st Qu.: 6.315
 Median : 9.0   Median : 9.0   Median : 9.0   Median : 8   Median : 7.580
 Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9   Mean   : 7.501
 3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8   3rd Qu.: 8.570
 Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19   Max.   :10.840
       y2              y3              y4
 Min.   :3.100   Min.   : 5.39   Min.   : 5.250
 1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170
 Median :8.140   Median : 7.11   Median : 7.040
 Mean   :7.501   Mean   : 7.50   Mean   : 7.501
 3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190
 Max.   :9.260   Max.   :12.74   Max.   :12.500
> cor(anscombe$x1,anscombe$y1)
[1] 0.8164205
> cor(anscombe$x2,anscombe$y2)
[1] 0.8162365
> data(beavers)
> names(beaver1)
[1] "day" "time" "temp" "activ"
> plot(beaver1)
```

```
anscombe    data.frame 8    2.4 KB   11 obs. of 8 variables
beaver1     data.frame 4    4.5 KB   114 obs. of 4 variables
beaver2     data.frame 4    4.1 KB   100 obs. of 4 variables
```

So, to load the Beavers data set, as usual type data beavers, and as you can see in the environment here, the work space beaver1 and beaver2 have been loaded. Once again, the type of data will be shown the moment you access the beaver1. So, for example, if I want to know what type it is, just start typing the name and R studio will show you that it is a data frame as well. So how many columns does the data frame beaver1 have? It has 4 columns. To know what are the names of those, once again we type… we look at the attributes or the names, sorry, of the columns, and it shows that the first two columns give me the day and time at which this record measurements were taken, and the third one is a temperature that’s of interest to us.

(Refer Slide Time: 10:47)



```
> ?datasets
> data("anscombe")
> anscombe$names
NULL
> names(anscombe)
[1] "x1" "x2" "x3" "x4" "y1" "y2" "y3" "y4"
> plot(anscombe$x1,anscombe$y1,pch=19,col='red')
> summary(anscombe)
       x1             x2             x3             x4           y1        
 Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8   Min.   : 4.260  
 1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8   1st Qu.: 6.315  
 Median : 9.0   Median : 9.0   Median : 9.0   Median : 8   Median : 7.580  
 Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9   Mean   : 7.501  
 3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8   3rd Qu.: 8.570  
 Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19   Max.   :10.840  
       y2              y3              y4        
 Min.   :3.100   Min.   : 5.39   Min.   : 5.250  
 1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170  
 Median :8.140   Median : 7.11   Median : 7.040  
 Mean   :7.501   Mean   : 7.50   Mean   : 7.501  
 3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190  
 Max.   :9.260   Max.   :12.74   Max.   :12.500  
> cor(anscombe$x1,anscombe$y1)
[1] 0.8164205
> cor(anscombe$x2,anscombe$y2)
[1] 0.8162365
> su
```

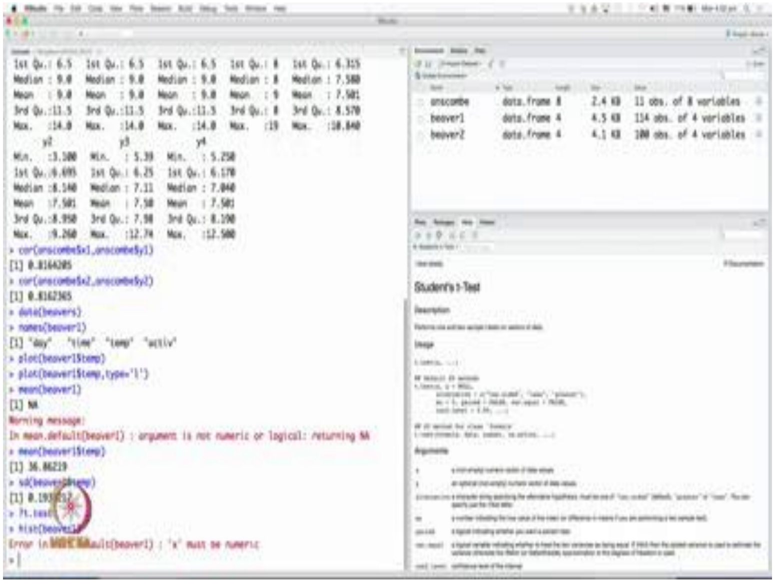
```
anscombe   data.frame 8   2.4 KB   11 obs. of 8 variables
beaver1    data.frame 4   4.5 KB   114 obs. of 4 variables
beaver2    data.frame 4   4.1 KB   100 obs. of 4 variables
```

Let’s plot the temperature. So, we say plot beaver1 dollar temperature. And when I do this, it produces a scatter plot, but I want ideally also line plot, and I can ask for a line plot if I want, and then this is a line plot. I can also ask for both lines and points and so on; the story is endless; but this is how the temperature series looks like over two days of recording. Now, as you can see, the temperature hovers around a 36-37 and so on and there is some variability to it. Now the question is - should I ask whether this data comes out of a deterministic or a stochastic process? This is univariate series. Suppose, I want to predict what will be the temperature of the beaver the following day; I want build a model, should I treat this as a deterministic process? Now, I can treat this as a deterministic process if I see, at least from a visual analysis, that a mathematical function can explain this nicely.

Now, of course, I can always fit a polynomial of 113th degree to explain the data perfectly, but unfortunately the 113th degree polynomial will fail miserably in predicting the next point. That also is an indication of lack of determinism in the data, and also, it do not have any other factors that affect the beaver’s temperature to help me predict this. So, it may be a wise thing to assume that this temperature series is stochastic, but on the other hand I also find some trends, and we will talk about it in the next and final data set that will take up. So, for now we can treat this data set to be stochastic and ask for mean of beaver. For example, I can ask what is the mean of, sorry, beaver1-dollar temperature alright; we can do that; and that is the average. We can also ask for the standard deviation. You can recall the previous commands in the history by using the up and down arrows and this is the standard deviation and so on.

Now, what we want to do is also compare the means; that is suppose my hypothesis is that these two beavers have the same average temperature right. Definitely the temperature observations that I have are just one of the many possible readings that I could have obtained. So, I am imagining that there is a population of readings for beaver1, and population of readings for beaver2, and I am assuming that both these populations have same averages. Let us say I want test that. Now, statistically that's a hypothesis test. If mu 1 is a true mean for beaver1, and mu 2 is a true mean for beaver2, and here conducting a hypothesis test that mu 1 equals mu 2 or mu 1 minus mu 2 is 0.

(Refer Slide Time: 14:10)

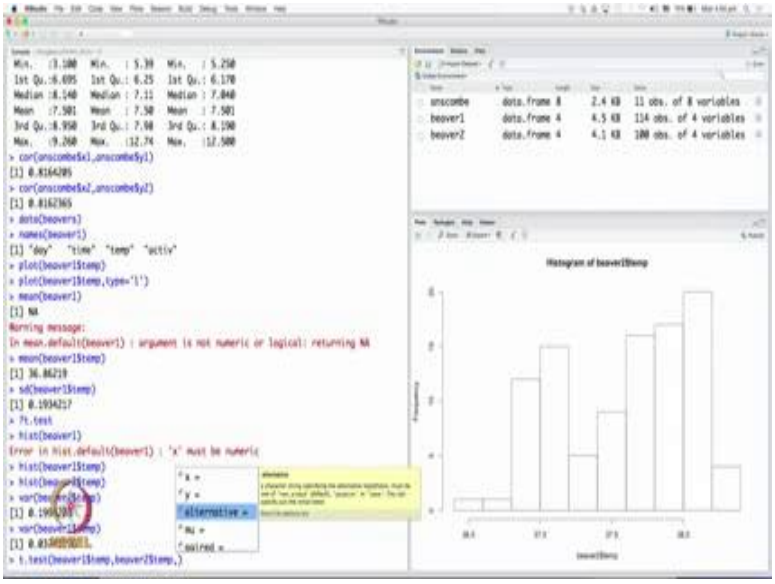


```
 1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8   1st Qu.: 6.315
 Median : 9.0   Median : 9.0   Median : 9.0   Median : 8   Median : 7.580
 Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9   Mean   : 7.501
 3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8   3rd Qu.: 8.570
 Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19   Max.   :10.840
       y2              y3              y4
 Min.   :3.100   Min.   : 5.39   Min.   : 5.250
 1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170
 Median :8.140   Median : 7.11   Median : 7.040
 Mean   :7.501   Mean   : 7.50   Mean   : 7.501
 3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190
 Max.   :9.260   Max.   :12.74   Max.   :12.500
> cor(anscombe$x1,anscombe$y1)
[1] 0.8164205
> cor(anscombe$x2,anscombe$y2)
[1] 0.8162365
> data(beavers)
> names(beaver1)
[1] "day"  "time"  "temp"  "activ"
> plot(beaver1$temp)
> plot(beaver1$temp,type='l')
> mean(beaver1)
[1] NA
Warning message:
In mean.default(beaver1) : argument is not numeric or logical: returning NA
> mean(beaver1$temp)
[1] 36.86219
> sd(beaver1$temp)
[1] 0.1934217
> ?t.test
> hist(beaver1)
Error in hist.default(beaver1) : 'x' must be numeric
> |
```

```
anscombe   data.frame 8   2.4 KB   11 obs. of 8 variables
beaver1    data.frame 4   4.5 KB   114 obs. of 4 variables
beaver2    data.frame 4   4.1 KB   100 obs. of 4 variables
```

Now, we are not going to go into the theory, but let me quickly tell you the routine or the command that helps you compare averages of two different populations, and that's the t dot test command, which is also known as a student's t test. The t refers to the distribution of the statistic that we shall use to carry out the hypothesis test. Remember statistic is some mathematical function of the data. So, what is done, at least theoretically, just to give you an idea, is that sample means of both the beaver1 and beaver2 series are computed, and statistically we look at the distance - you can say so crudely - between the sample means, in presence of the fact that they have come from two different populations. What is a difference between them? I am hypothesizing that the means are the same, then, what is a difference? The difference is in the variability, in the variances; that is, they come out of different spreads. To know that, for example, whether my assumption that different variabilities hold good, we can look at a histogram and look at the spread also. So, let’s look at the histogram of beaver1 which gives me an idea of the distribution of the data.

(Refer Slide Time: 15:25)

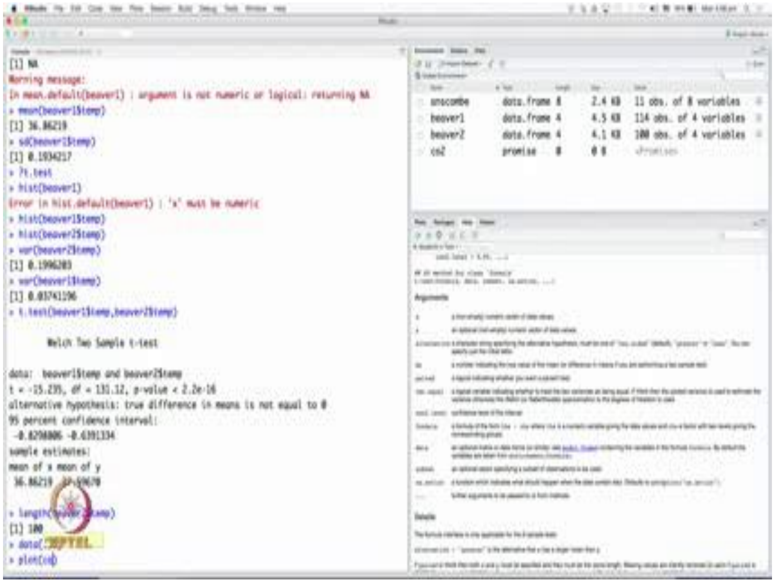


```
 Min.   :3.100   Min.   : 5.39   Min.   : 5.250
 1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170
 Median :8.140   Median : 7.11   Median : 7.040
 Mean   :7.501   Mean   : 7.50   Mean   : 7.501
 3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190
 Max.   :9.260   Max.   :12.74   Max.   :12.500
> cor(anscombe$x1,anscombe$y1)
[1] 0.8164205
> cor(anscombe$x2,anscombe$y2)
[1] 0.8162365
> data(beavers)
> names(beaver1)
[1] "day"   "time"  "temp"  "activ"
> plot(beaver1$temp)
> plot(beaver1$temp,type='l')
> mean(beaver1)
[1] NA
Warning message:
In mean.default(beaver1) : argument is not numeric or logical: returning NA
> mean(beaver1$temp)
[1] 36.86219
> sd(beaver1$temp)
[1] 0.1934217
> ?t.test
> hist(beaver1)
Error in hist.default(beaver1) : 'x' must be numeric
> hist(beaver1$temp)
> hist(beaver2$temp)
> var(beaver2$temp)
[1] 0.19[illegible]
> var(beaver1$temp)
[1] 0.03[illegible]
> t.test(beaver1$temp,beaver2$temp,)
```

```
anscombe   data.frame 8   2.4 KB   11 obs. of 8 variables
beaver1    data.frame 4   4.5 KB   114 obs. of 4 variables
beaver2    data.frame 4   4.1 KB   100 obs. of 4 variables
```

So, now on the plot you see the histogram of data, it is beaver1 - temperature histogram - indicating some kind of a Gaussian distribution with the mean being the value that we have calculated here. What about beaver2? Does it look like a Gaussian distribution or some other distribution? Now, it doesn’t look exactly like a Gaussian; I am not even so close, but what we can see strikingly is that the variability; if I look at the variance of beaver2 verses variance of beaver1, the variabilities - these are sample variabilities; that is, these have been computed from data - they are quite different from each other. So, there is some justification to the assumption that we are making that is the temperature readings of these two beavers have come from different variability. In other words, the variability of temperature in both these beavers are quite different from each other. And it’s possible even among human beings temperatures can vary quite significantly, but the means can be the same. So, we can now test the hypothesis and to know the syntax of any routine, for example, I can press the tab, and it tells me what are the default ones. I have to supply the first series, and then, supply the second series. And then also, there are other options that you want - what is the alternative hypothesis? Whenever I am testing an hypothesis or any null hypothesis there is an alternative hypothesis, then I have to specify. Here the alternative hypothesis is that the means are different from each other.

(Refer Slide Time: 17:31)



```
[1] NA
Warning message:
In mean.default(beaver1) : argument is not numeric or logical: returning NA
> mean(beaver1$temp)
[1] 36.86219
> sd(beaver1$temp)
[1] 0.1934217
> ?t.test
> hist(beaver1)
Error in hist.default(beaver1) : 'x' must be numeric
> hist(beaver1$temp)
> hist(beaver2$temp)
> var(beaver2$temp)
[1] 0.1996203
> var(beaver1$temp)
[1] 0.03741196
> t.test(beaver1$temp,beaver2$temp)

        Welch Two Sample t-test

data:  beaver1$temp and beaver2$temp
t = -15.235, df = 131.12, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8298806 -0.6391334
sample estimates:
mean of x mean of y
 36.86219  37.59670

> length(beaver2$temp)
[1] 100
> data(
> plot(cs)
```

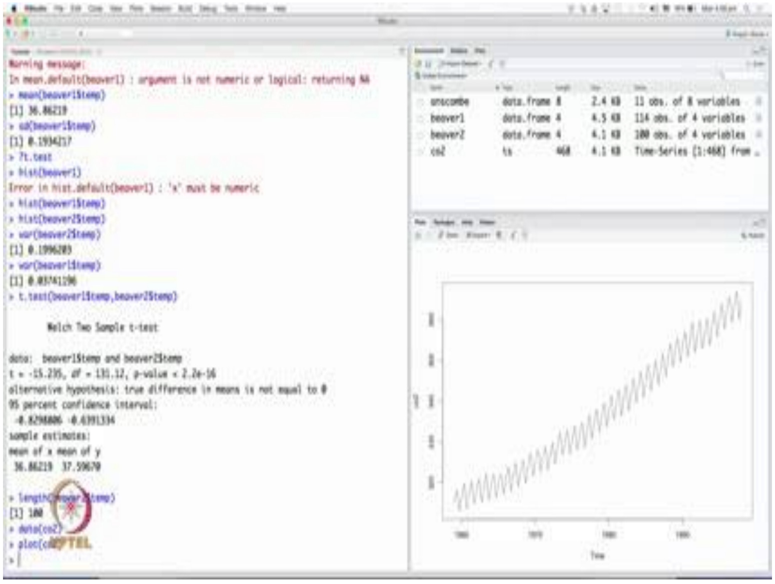
anscombe	data.frame	8	2.4 KB	11 obs. of 8 variables
beaver1	data.frame	4	4.5 KB	114 obs. of 4 variables
beaver2	data.frame	4	4.1 KB	100 obs. of 4 variables
cs2	promise	0	0 B	<Promise>

In other situations, I may test the mean of beaver1 or the true average of a temperature of beaver1 is greater than that of beaver2 and so on, but we are not interested in that here. So, the default alternative if you look up the t test - t dot test - see documentation it says… it gives you the default values which is for the alternative one it says they are not different and there are many other options. For example, variance being equal - am I assuming the variability to be equal, no, and that’s a default option as well. It says false. So, I don’t have to do anything with that. And there are a few other options which I am not going to explain now, it requires a bit of theory, but let’s do this.

Let’s see what the t dot test tells me. It reports a bunch of things, but what is of importance is this p value alright. And of course, it gives you the means of x and y. When you look at the sample means they look pretty close. I mean Of course, in the sense you may say no, I may say yes, and that’s the reason why we are performing a statistical analysis; but the statistical analysis with its very, very low P value is indicating that the hypothesis is… null hypothesis is the means are equal does not hold good; it has to be rejected. Whenever the p value is low, the null hypothesis must go; so, that is a nice phrase that you will find in Ogunnaike’s book. It is also telling you the alternative hypothesis against which the null hypothesis has been tested, and it also gives you the 95 percent confidence interval. The 95 confidence interval is on something. What is that? It is on the difference in the means. So, what it is testing is that the means are not different from each other, that’s null hypothesis. And if you turn to statistics books, they will tell you that a hypothesis test of the form mu 1 minus mu 2 equals 0 can be also conducted by looking at the confidence interval for the difference in means.

So, the confidence interval for the difference in means is reported here, and it does not include the postulated value which is 0. If the confidence interval had included the postulated value for the difference in true means which is 0, then we cannot reject the null hypothesis. There is not enough evidence in the data to reject the null hypothesis. In this case, we have sufficient evidence in the data to come to a conclusion that the temperature - average temperature - of beaver1 and average temperature of beaver2 are different from each other. And we have done this despite the fact that the readings in beaver1 and beaver2 are of different sizes. Beaver1 has 114 data points; beaver2 has 100 data points. How do I know that? I can ask for the length or I can look the help and you can see here 100 data points; where as we had 114 data points. So, the theory allows you to collect different sample sizes, but the question is whether this test itself has been conducted correctly. Theoretically does it satisfy theoretical assumptions? Probably not because this t test actually assumes that both the samples have comes from Gaussian distributions, and we have seen that the beaver2 temperature series is not really conforming to the Gaussian distribution, but may be had we collected larger and larger samples, then the assumptions might have been met more strictly alright. So, the point to keep in mind is not just reporting the analysis, but also the assumptions that we have made and whether the data has met those assumptions.

(Refer Slide Time: 21:32)



```
Warning message:
In mean.default(beaver1) : argument is not numeric or logical: returning NA
> mean(beaver1$temp)
[1] 36.86219
> sd(beaver1$temp)
[1] 0.1934217
> ?t.test
> hist(beaver1)
Error in hist.default(beaver1) : 'x' must be numeric
> hist(beaver1$temp)
> hist(beaver2$temp)
> var(beaver2$temp)
[1] 0.1996283
> var(beaver1$temp)
[1] 0.03741296
> t.test(beaver1$temp,beaver2$temp)

        Welch Two Sample t-test

data:  beaver1$temp and beaver2$temp
t = -15.235, df = 131.12, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8298806 -0.6391334
sample estimates:
mean of x mean of y
 36.86219  37.59670

> length(beaver2$temp)
[1] 100
> data(co2)
> plot(co2)
>
```

```
anscombe   data.frame 8     2.4 KB   11 obs. of 8 variables
beaver1    data.frame 4     4.5 KB   114 obs. of 4 variables
beaver2    data.frame 4     4.1 KB   100 obs. of 4 variables
co2        ts         468   4.1 KB   Time-Series [1:468] from ...
```

The final thing that I want to show you is another series, which is the carbon dioxide emissions in a certain region, during a certain period. Again, I welcome you to look up the help on this data set. Just want to show you the data set. And here, I have loaded the carbon dioxide data set and we shall plot this series. And you can see in this series there is a trend, the x axis is time - the year in which it was the carbon dioxide emission was collected. There is a linear trend or probably a slightly parabolic trend, we do not know, but vividly there are two things - a linear trend and then there is an oscillatory nature to it. Now, once again the question - should I treat this as a deterministic process or a stochastic process? On the face of it, it appears deterministic, because it's so dominant, there is a mathematical function that I can fit to explain the trend. I can also explain the oscillations using sinusoids. I can determine the frequency.

How do I determine the frequency? By looking at the periodogram or the power spectral density. You can fit a linear trend. I will show you how to fit linear trends when we go to the lecture on modeling skills. Basically, it is a matter of a regression, but this shows you that there is a possibility that you can think of the series as deterministic. However, once you have removed the deterministic part, you should ask if there is anything left in the series to be explained, and if that has some irregularities, some stochastic nature to it, then you come to the conclusion that the series is a mix of deterministic and stochastic processes. Here, the determinism is purely as a function of time. In many processes, the deterministic nature can come about by a certain cause that you already know right. In the reactor temperature example that we discussed, if I give you the coolant flow and the reactor temperature series, using the coolant flow you can explain the variations in the reactor temperature significantly; yet there will be something left in the temperature that you may not be able to explain using a coolant flow which will be due to the sensor noise; there you have a mix of deterministic and stochastic process.We are not talking about linear or non-linear here.

Anyway, so hopefully I have introduced to you through these examples first of all the great free-open source software package called R, because some of the students have seen in the forum have requested for introduction to, an exposure to some programming language, but what is needed in data analysis is not necessary programming language, but a nice software tool that confirms to the theory, and R has been written by many of the pioneers in the world of statistical data analysis.Therefore, there is more credibility to it and there is a nice user forum.You can ask any questions and it’s continuously developed. R studio is a fantastic piece of GUI for R. So, enjoy and yeah write to us if you have any questions. Hopefully you enjoyed the lecture.

Thank you.