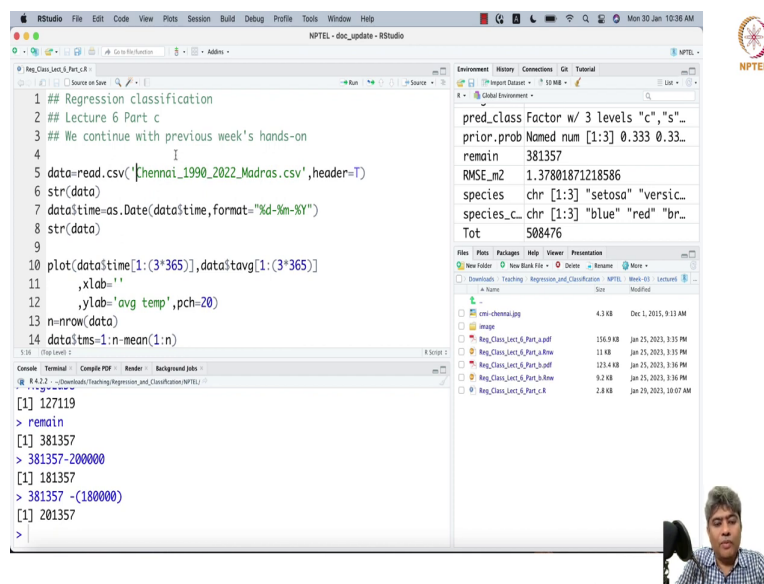


Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Chennai Mathematical Institute

Lecture - 22
Hands on with R Part - 5

Welcome back to last part of lecture 6. And in this part, we will do some Hands on with R. And let me just you know try to share with this stuff.

(Refer Slide Time: 00:33)



The screenshot displays the RStudio interface. The script editor on the left contains the following R code:

```
1 ## Regression classification
2 ## Lecture 6 Part c
3 ## We continue with previous week's hands-on
4
5 data=read.csv('Chennai_1990_2022_Madras.csv',header=T)
6 str(data)
7 dateTime=as.Date(dateTime,format="%d-%m-%Y")
8 str(dateTime)
9
10 plot(dateTime[1:(3*365)],data$avg[1:(3*365)])
11       ,xlab=''
12       ,ylab='avg temp',pch=20)
13 n=nrow(data)
14 data$ms=1:n-mean(1:n)
```

The console on the left shows the output of the `str(data)` command:

```
[1] 127119
> remain
[1] 381357
> 381357-200000
[1] 181357
> 381357 -(180000)
[1] 201357
>
```

The environment pane on the right shows the following objects:

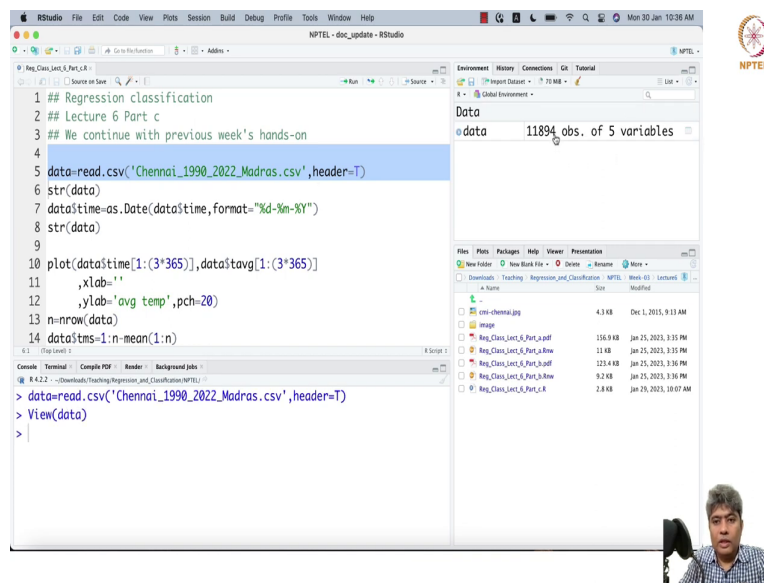
Object	Class	Attributes
pred_class	Factor w/ 3 levels "c", "s", "r"	
prior_prob	Named num [1:3]	0.333 0.333 0.333
remain	dbl	381357
RMSE_m2	dbl	1.37801871218586
species	chr [1:3]	"setosa" "versicol" "virginica"
species_c	chr [1:3]	"blue" "red" "br"
Tot	dbl	508476

The Files pane on the right shows a list of files in the current directory, including `chennai.jpg`, `image`, and several PDF files related to the course.

So, this is and this is the quote that I already have shared with you in the lecture material part that we are using Chennai, 1990 we have done some work with the data, this data set. And in this data set, what we have done is we have we tried to fit simple sine cosine kind of I mean some regression model with sine cosine Fourier engine term or engineer term.

So, let us start with this. We have this data. Say, let me just first clean the environment, that will help you.

(Refer Slide Time: 01:30)




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


```
1 ## Regression classification
2 ## Lecture 6 Part c
3 ## We continue with previous week's hands-on
4
5 data=read.csv('Chennai_1990_2022_Madras.csv',header=T)
6 str(data)
7 data$time=as.Date(data$time,format="%d-%m-%Y")
8 str(data)
9
10 plot(data$time[1:(3*365)],data$avg[1:(3*365)])
11     ,xlab=''
12     ,ylab='avg temp',pch=20)
13 n=nrow(data)
14 data$time=1:n-mean(1:n)
```

The console window shows the execution of the code:

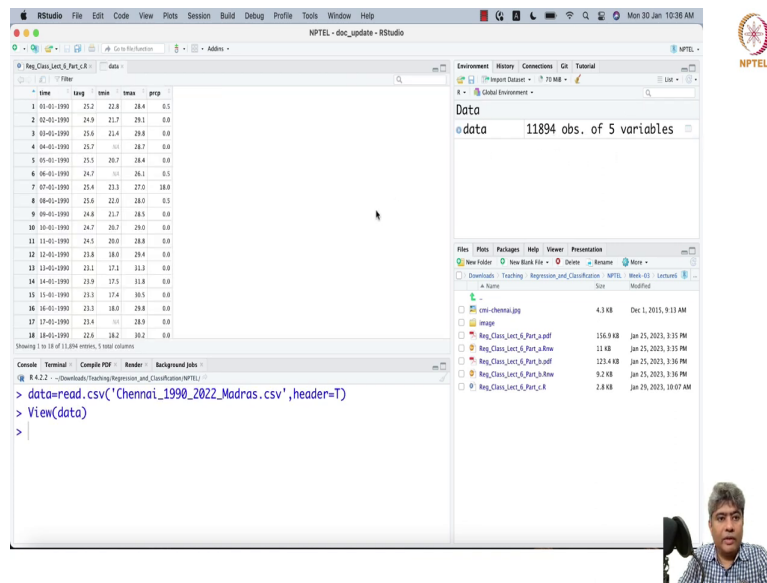
```
> data=read.csv('Chennai_1990_2022_Madras.csv',header=T)
> View(data)
>
```

The Environment pane on the right shows a data object named 'data' with 11894 observations and 5 variables. The Files pane on the right shows a list of files in the current directory, including 'csi-chemical.jpg', 'image', and several PDF files related to the course.





(Refer Slide Time: 01:33)



The screenshot displays the RStudio interface. The main window shows a data frame with 11894 observations and 5 variables. The variables are 'time', 'temp', 'minw', 'maxw', and 'prop'. The console shows the following code:

```
> data=read.csv("Chennai_1990_2022_Madras.csv",header=T)
> View(data)
>
```

The data frame contains the following data:

	time	temp	minw	maxw	prop
1	01-01-1990	25.2	22.8	28.4	0.5
2	02-01-1990	24.9	21.7	29.1	0.0
3	03-01-1990	25.6	21.4	29.8	0.0
4	04-01-1990	25.7	NA	28.7	0.0
5	05-01-1990	25.5	20.7	28.4	0.0
6	06-01-1990	24.2	NA	26.1	0.5
7	07-01-1990	25.4	21.3	27.0	18.0
8	08-01-1990	25.6	22.0	28.0	0.5
9	09-01-1990	24.8	21.7	28.5	0.0
10	10-01-1990	24.7	20.7	29.0	0.0
11	11-01-1990	24.5	20.0	28.8	0.0
12	12-01-1990	23.8	18.0	29.4	0.0
13	13-01-1990	23.1	17.1	31.3	0.0
14	14-01-1990	23.9	17.5	31.8	0.0
15	15-01-1990	23.3	17.4	30.5	0.0
16	16-01-1990	23.3	18.0	29.8	0.0
17	17-01-1990	23.4	NA	28.9	0.0
18	18-01-1990	23.8	18.2	30.7	0.0

So, first, so this is the data set that we have. We have seen this before, last week.

(Refer Slide Time: 01:41)

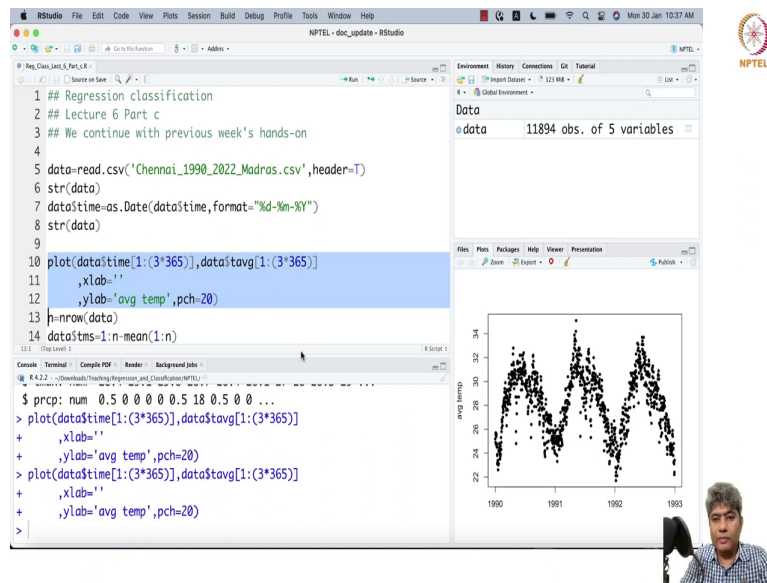
```
1 ## Regression classification
2 ## Lecture 6 Part c
3 ## We continue with previous week's hands-on
4
5 data=read.csv('Chennai_1990_2022_Madras.csv',header=T)
6 str(data)
7 data$time=as.Date(data$time,format="%d-%m-%Y")
8 str(data)
9
10 plot(data$time[1:(3*365)],data$tavg[1:(3*365)])
11     ,xlab=''
12     ,ylab='avg temp',pch=20)
13 n=nrow(data)
14 data$tns=1:n-mean(1:n)
```

```
> str(data)
'data.frame': 11894 obs. of 5 variables:
 $ time: chr  "01-01-1990" "02-01-1990" "03-01-1990" "04-01-1990" ...
 $ tavg: num  25.2 24.9 25.6 25.7 25.5 24.7 25.4 25.6 24.8 24.7 ...
 $ tmi:  num  22.8 21.7 21.4 NA 20.7 NA 23.3 22 21.7 20.7 ...
 $ tmax: num  28.4 29.1 29.8 28.7 28.4 26.1 27 28 28.5 29 ...
 $ prcp: num  0.5 0 0 0 0 0.5 18 0.5 0 0 ...
```

The screenshot also shows the RStudio interface with the Environment pane displaying 'data' with 11894 observations and 5 variables. The Files pane shows a list of files including 'chennai-1990-2022-madras.csv' and various lecture files.

Now, the structure of the data set is everything is numeric but time was taken as character. So, we convert it into time and now you have this as a date format.

(Refer Slide Time: 01:58)



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

```
1 ## Regression classification
2 ## Lecture 6 Part c
3 ## We continue with previous week's hands-on
4
5 data=read.csv('Chennai_1990_2022_Madras.csv',header=T)
6 str(data)
7 data$time.as.Date(data$time,format="%d-%m-%Y")
8 str(data)
9
10 plot(data$time[1:(3*365)],data$avg[1:(3*365)])
11     ,xlab=''
12     ,ylab='avg temp',pch=20)
13 #nrow(data)
14 data$ms=1:n-mean(1:n)
```

The console window shows the execution of the code, including the output of `str(data)` and the execution of the `plot` function. The plot window displays a scatter plot of average temperature (avg temp) over time, with the x-axis ranging from 1990 to 1993 and the y-axis ranging from 22 to 32. The plot shows a clear seasonal pattern with peaks in the summer months and troughs in the winter months.

The Data pane on the right shows the structure of the data:

```
Data
data 11894 obs. of 5 variables
```

The Environment pane shows the current environment with 11894 observations and 5 variables.

The Files pane shows the current directory structure.

The Packages pane shows the installed packages.

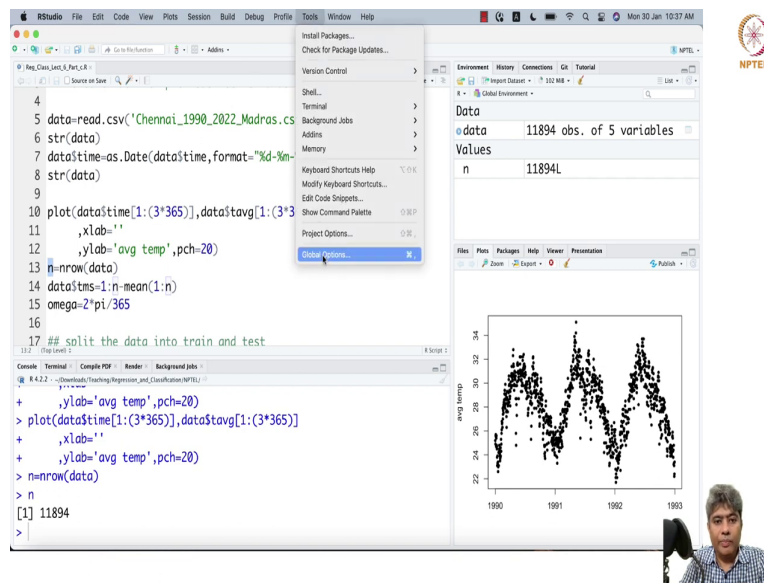
The Help pane shows the help documentation for the current function.

The Console window shows the following output:

```
$ prcp: num 0.5 0 0 0 0 0.5 18 0.5 0 ...
> plot(data$time[1:(3*365)],data$avg[1:(3*365)])
+ ,xlab=''
+ ,ylab='avg temp',pch=20)
> plot(data$time[1:(3*365)],data$avg[1:(3*365)])
+ ,xlab=''
+ ,ylab='avg temp',pch=20)
>
```

Then, now we plot this. Now, it is the data set. We have 118895 data points.

(Refer Slide Time: 02:11)



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

```
4  
5 data=read.csv('Chennai_1990_2022_Madras.csv')  
6 str(data)  
7 data$time=as.Date(data$time,format='%d-%m-%Y')  
8 str(data)  
9  
10 plot(data$time[1:(3*365)],data$avg[1:(3*365)])  
11     ,ylab='avg temp',pch=20  
12  
13 n=nrow(data)  
14 data$time=1:n-mean(1:n)  
15 omega=2*pi/365  
16  
17 ## split the data into train and test
```

The console shows the execution of the plotting code:

```
+ ,ylab='avg temp',pch=20  
> plot(data$time[1:(3*365)],data$avg[1:(3*365)])  
+ ,ylab='avg temp',pch=20  
> n=nrow(data)  
> n  
[1] 11894  
>
```

The Environment pane shows the 'data' object with 11894 observations and 5 variables. The plot shows 'avg temp' on the y-axis (ranging from 22 to 28) against time on the x-axis (ranging from 1990 to 1993). The plot displays a dense cloud of points forming a seasonal wave pattern. The NPTEL logo is visible in the top right corner of the RStudio window.

Let me increase the font size a bit. You go to tools, you go to global options, then appearance (Refer Time: 02:25) maybe zoom 200 percent and apply and ok, right.

(Refer Slide Time: 02:21)

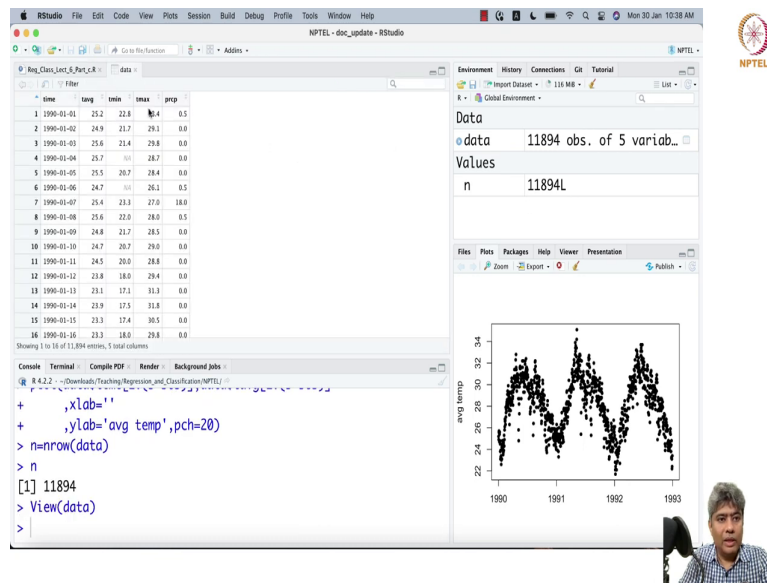
The screenshot displays the RStudio interface with the following components:

- Code Editor:** Contains R code for reading a CSV file, converting it to a Date object, and plotting it. The code includes comments and function definitions for handling missing values and labels.
- Console:** Shows the execution of the code, resulting in the output: `[1] 11894`.
- Environment:** Shows a data object with 11894 observations and 5 variables.
- Plot:** A scatter plot showing data points over time from 1990 to 1993. The plot shows a clear seasonal pattern with peaks and troughs.
- Options Dialog:** A dialog box is open, showing the 'Editor theme' set to 'Students' and the 'Editor font size' set to 12. The 'Apply' button is highlighted.

The R code in the editor is as follows:

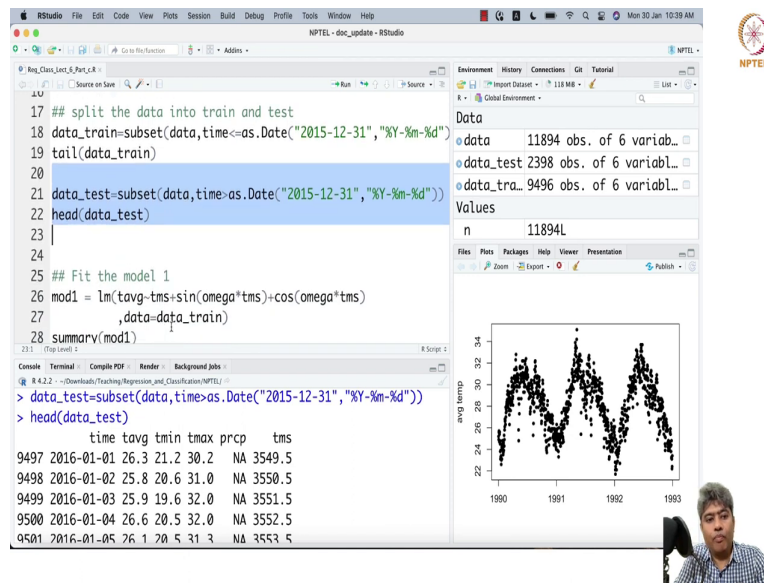
```
4
5 data=read.csv("Chennai_1990_2022_Medias_soul_header.T")
6 str(data)
7 data$time=as.Date(data$time,format="%Y-%m-%d")
8 str(data)
9
10 plot(data$time[1:(3*365)],data$avgtemp,
11       ,ylab="avg temp",pch=20)
12
13 n=nrow(data)
14 data$time=1:n-mean(1:n)
15 omega=2*pi/365
16
17 ## split the data into train and test data
18
19 # define the plot function
20 plot <- function(x,y) {
21   # plotting of R objects
22   if (is.function(x)) {
23     if (is.null(attr("ylab"))) {
24       if (missing(y)) {
25         y <- NULL
26       }
27     }
28     # check for ylab
29     hasylab <- function(x) {
30       all(is.na(x$names)) && !is.null(x$names)
31     }
32     if (hasylab(x)) {
33       plot.function(x,y)
34     } else {
35       plot(x,y)
36     }
37   }
38 }
```

(Refer Slide Time: 02:42)



And then what we are doing, we are creating another data set another column. So, there is a time t , average temperature, minimum temperature, maximum temperature and precipitation, right.

(Refer Slide Time: 03:14)



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

```
17 ## split the data into train and test
18 data_train=subset(data,time<=as.Date("2015-12-31", "%Y-%m-%d"))
19 tail(data_train)
20
21 data_test=subset(data,time>as.Date("2015-12-31", "%Y-%m-%d"))
22 head(data_test)
23
24
25 ## Fit the model 1
26 mod1 = lm(tavg-tms+sin(omega*tms)+cos(omega*tms)
27           ,data=data_train)
28 summary(mod1)
```

The console output shows the execution of the code:

```
> data_test=subset(data,time>as.Date("2015-12-31", "%Y-%m-%d"))
> head(data_test)
      time tavg tmin tmax prcp  tms
9497 2016-01-01 26.3 21.2 30.2  NA 3549.5
9498 2016-01-02 25.8 20.6 31.0  NA 3550.5
9499 2016-01-03 25.9 19.6 32.0  NA 3551.5
9500 2016-01-04 26.6 20.5 32.0  NA 3552.5
9501 2016-01-05 26.1 20.5 31.3  NA 3553.5
```

The environment pane shows the following data objects:

Object	Observations	Variables
data	11894	6
data_test	2398	6
data_train	9496	6

The plot shows the average temperature (avg temp) over time from 1980 to 1993. The y-axis ranges from 22 to 34, and the x-axis shows years from 1980 to 1993. The plot shows a clear seasonal cycle with peaks around 32-34 and troughs around 22-24.

Now, I am creating a new column omega, ok. Now, after creating column, you can see that what we have done. I have subtract the average time; just location, I just shift the location and then I split the data into train and test. So, any data before 31st December 2015, I have used it as train data.

And if you just put a tail, so you can see the last value is of the train data is 31st December 2015. And any data before after 31st December 2015 is test data. So, I am just plotting the first few rows. So, from 1st January 2016, we are going to use it as test data. So, then we fit first model, ok.

(Refer Slide Time: 04:06)

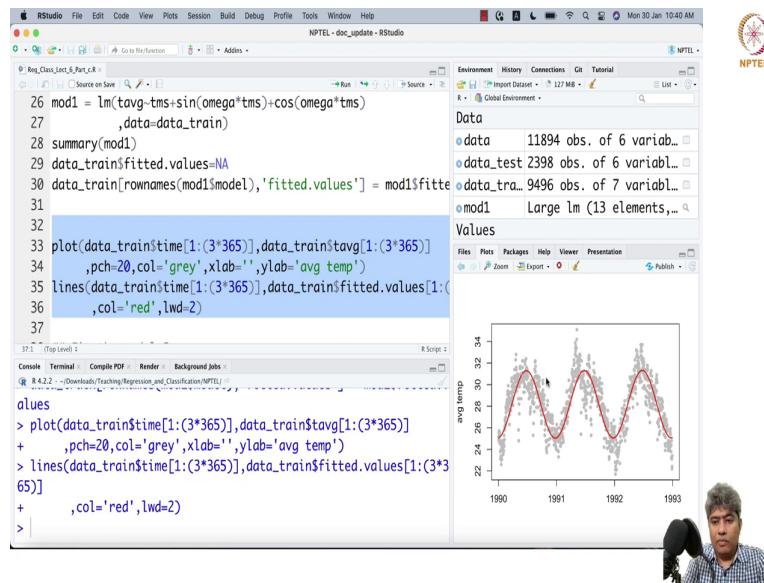
The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data inspection and model fitting:

```
22 head(data_test)
23
24
25 ## Fit the model 1
26 mod1 = lm(tavg~tms+sin(omega*tms)+cos(omega*tms)
27           ,data=data_train)
28 summary(mod1)
29 data_train$fitted.values=NA
30 data_train[rownames(mod1$model),'fitted.values'] = mod1$fitted.values
31
32
33 plot(data_train$time[1:(3*365)],data_train$tavg[1:(3*365)])
```
- Environment:** Lists objects in the workspace:
 - data: 11894 obs. of 6 variables
 - data_test: 2398 obs. of 6 variables
 - data_tra: 9496 obs. of 6 variables
 - mod1: Large lm (13 elements)
- Console:** Shows the execution of the code and the output of the `summary(mod1)` function:

```
9499 2016-01-03 25.9 19.6 32.0 NA 3551.5
9500 2016-01-04 26.6 20.5 32.0 NA 3552.5
9501 2016-01-05 26.1 20.5 31.3 NA 3553.5
9502 2016-01-06 26.4 20.5 31.0 NA 3554.5
> mod1 = lm(tavg~tms+sin(omega*tms)+cos(omega*tms)
+           ,data=data_train)
>
```
- Plots:** A scatter plot titled 'avg temp' showing the relationship between time (x-axis, 1990-1993) and average temperature (y-axis, 22-34). The plot shows a clear seasonal oscillation.
- NPTEL Logo:** Located in the top right corner of the RStudio window.
- Speaker:** A small video feed of a person is visible in the bottom right corner of the RStudio window.

(Refer Slide Time: 04:30)



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

```
26 mod1 = lm(tavg-tms*sin(omega*tms)+cos(omega*tms)
27 ,data=data_train)
28 summary(mod1)
29 data_train$fitted.values=NA
30 data_train[rownames(mod1$model),'fitted.values'] = mod1$fitted.values
31
32
33 plot(data_train$time[1:(3*365)],data_train$tavg[1:(3*365)])
34 ,pch=20,col='grey',xlab='',ylab='avg temp')
35 lines(data_train$time[1:(3*365)],data_train$fitted.values[1:(3*365)])
36 ,col='red',lwd=2)
37
```

The Environment pane on the right shows the following objects:

- data: 11894 obs. of 6 variables
- data_test: 2398 obs. of 6 variables
- data_tra: 9496 obs. of 7 variables
- mod1: Large lm (13 elements)

The Console window shows the execution of the following commands:

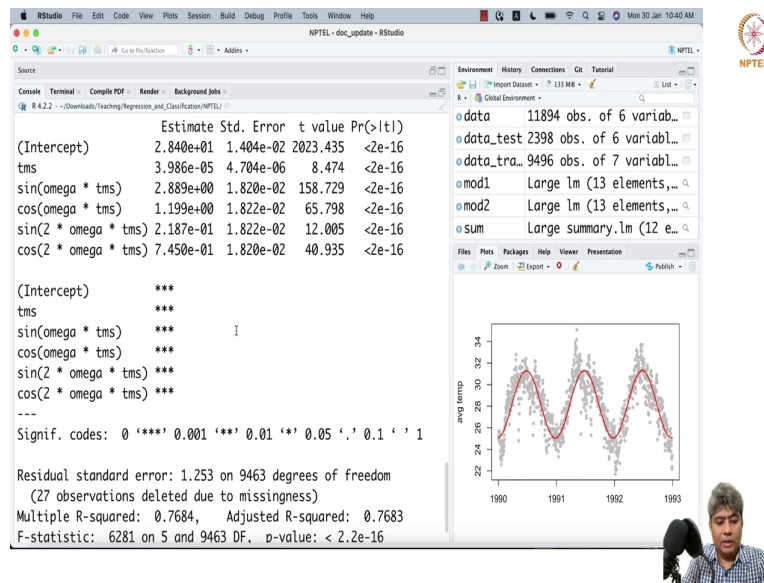
```
> plot(data_train$time[1:(3*365)],data_train$tavg[1:(3*365)])
+ ,pch=20,col='grey',xlab='',ylab='avg temp')
> lines(data_train$time[1:(3*365)],data_train$fitted.values[1:(3*365)])
+ ,col='red',lwd=2)
>
```

The plot window displays a scatter plot of average temperature (avg temp) over time, with a red line representing the fitted model. The x-axis ranges from 1980 to 1993, and the y-axis ranges from 22 to 34. The plot shows a clear seasonal pattern with three peaks and three troughs.

The NPTEL logo is visible in the top right corner of the RStudio window.

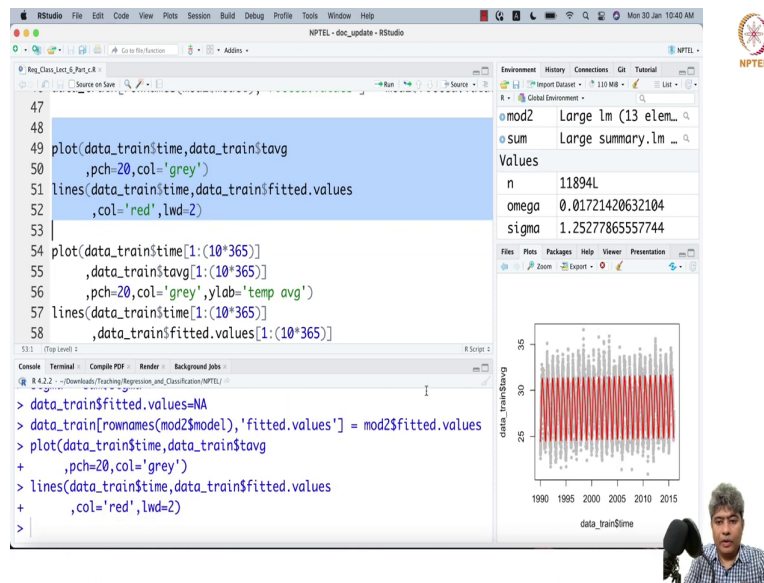
And this is the first model that we fit it. You can see with one that we did last week this model we fitted last week. Then, we created took the fitted model, fitted values and plot the fitted values through the average temperature.

(Refer Slide Time: 04:46)



Then, we fit the second model, ok. And what we are seeing that what we are seen the second model with sin 2 omega type t plus cos 2 omega t, they are all significant and name various fitted model, fit the model.

(Refer Slide Time: 05:17)

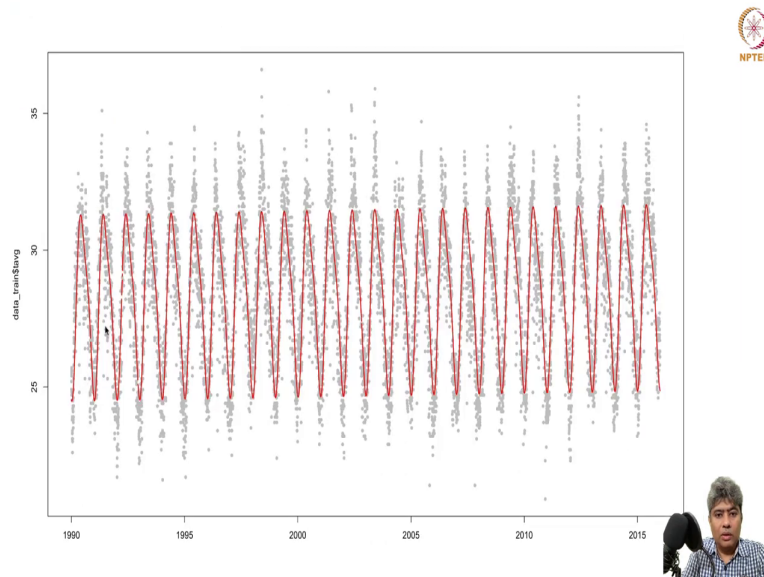


The image shows the RStudio interface with the following components:

- Source Editor:** Contains R code for plotting data and a fitted model. Lines 49-53 plot the full dataset, and lines 54-58 plot a subset of the data.
- Environment:** Shows the 'mod2' object as a 'Large lm (13 elem...)' and the 'sum' object as a 'Large summary.lm...'. The 'Values' section lists: n: 11894L, omega: 0.01721420632104, sigma: 1.25277865557744.
- Console:** Shows the execution of the following commands:

```
> data_train$fitted.values=NA
> data_train[rownames(mod2$model), 'fitted.values'] = mod2$fitted.values
> plot(data_train$time, data_train$avg
+       ,pch=20, col='grey')
> lines(data_train$time, data_train$fitted.values
+       , col='red', lwd=2)
>
```
- Plot:** A scatter plot of 'data_train\$avg' (y-axis, 25-35) versus 'data_train\$time' (x-axis, 1990-2015). The plot shows grey points with a red line representing the fitted model.
- NPTEL Logo:** Located in the top right corner.
- Video Feed:** A small video feed of a person is visible in the bottom right corner.

(Refer Slide Time: 05:24)



Then, so this is sort of a let me just know. So, this is you can; so, every year we have some kind of sine cosine behavior. This is kind of expected. At the same time, it is kind of going up. We can see there is an increasing trend, model is picking up a increasing trend.

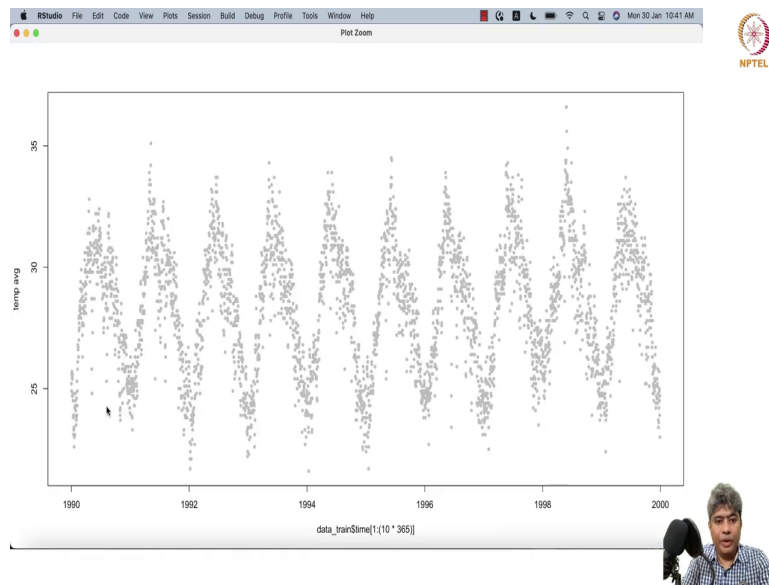
(Refer Slide Time: 05:50)

The screenshot displays the RStudio interface with the following components:

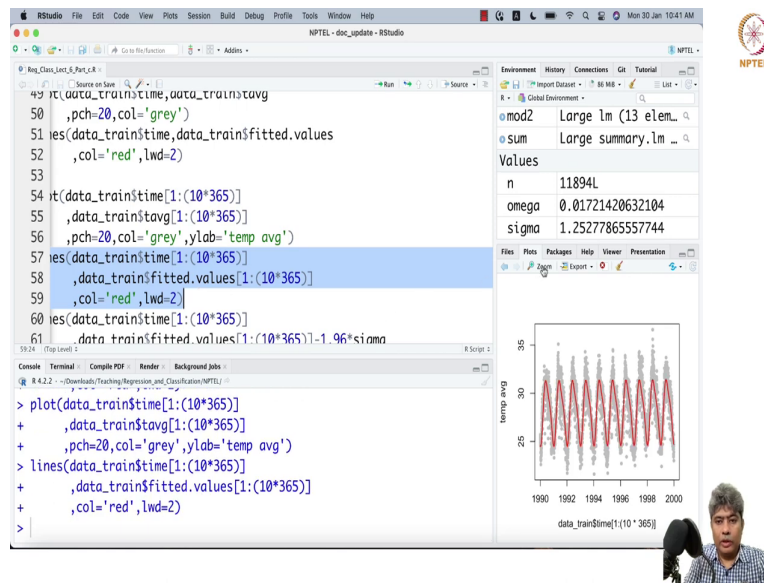
- Source Editor:** Contains R code for plotting time series data. Lines 49-53 show a plot of `data_train$time` and `data_train$avg` with a fitted line. Lines 54-59 show a zoomed-in plot of the same data.
- Environment:** Shows the current environment with variables `mod2` (Large lm (13 elem...)) and `sum` (Large summary, lm ...).
- Values:** Displays the values for the variables: `n` (11894L), `omega` (0.01721420632104), and `sigma` (1.25277865557744).
- Console:** Shows the execution of the R code, including the plot commands and the resulting output.
- Plot:** A time series plot showing `temp avg` on the y-axis (ranging from 25 to 35) and `data_train$time[1:(10*365)]` on the x-axis (ranging from 1990 to 2000). The plot displays a noisy time series with a fitted line.

The NPTEL logo is visible in the top right corner of the RStudio window.

(Refer Slide Time: 05:53)



(Refer Slide Time: 06:04)



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

```
49 plot(data_train$time, data_train$avg
50 ,pch=20,col='grey')
51 lines(data_train$time,data_train$fitted.values
52 ,col='red',lwd=2)
53
54 plot(data_train$time[1:(10*365)]
55 ,data_train$avg[1:(10*365)]
56 ,pch=20,col='grey',ylab='temp avg')
57 lines(data_train$time[1:(10*365)]
58 ,data_train$fitted.values[1:(10*365)]
59 ,col='red',lwd=2)
60 lines(data_train$time[1:(10*365)]
61 ,data_train$fitted.values[1:(10*365)]-1.96*sigma
```

The console window shows the execution of the following commands:

```
> plot(data_train$time[1:(10*365)]
+ ,data_train$avg[1:(10*365)]
+ ,pch=20,col='grey',ylab='temp avg')
+ lines(data_train$time[1:(10*365)]
+ ,data_train$fitted.values[1:(10*365)]
+ ,col='red',lwd=2)
+ 
```

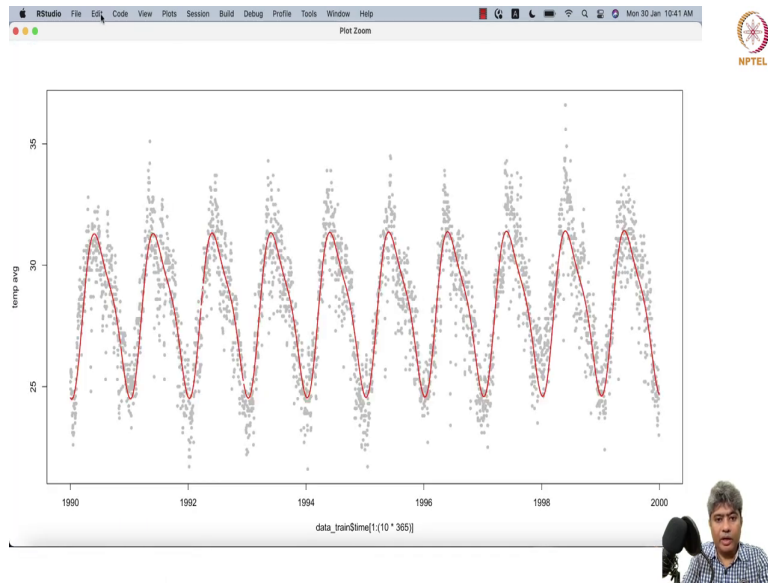
The Environment pane on the right shows the following values:

Variable	Value
n	11894L
omega	0.01721420632104
sigma	1.25277865557744

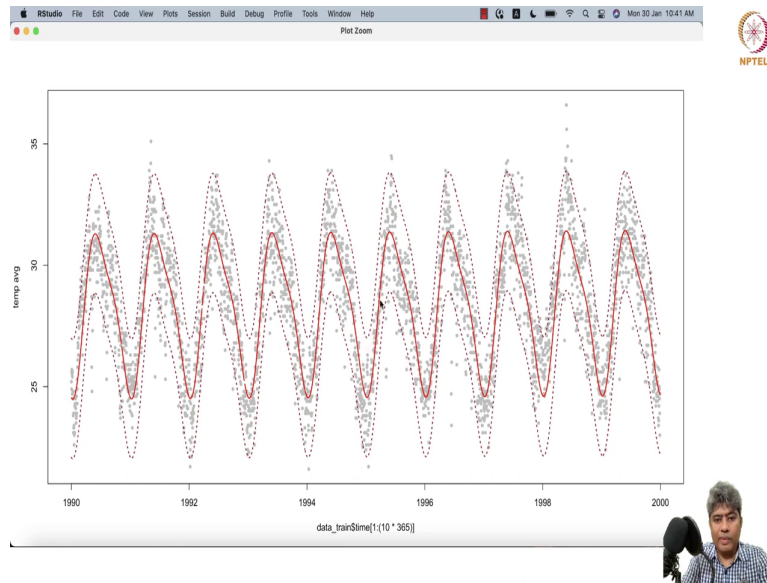
The plot on the right shows a time series of temperature data (temp avg) from 1990 to 2000. The y-axis ranges from 25 to 35. The data points are grey circles, and the fitted values are red lines. The plot shows a clear seasonal pattern with a peak around 30 and a trough around 25.

And we just here we just plotted the first 10 years from 1990 to 2000. And so, here is the first 10 years average expected behavior.

(Refer Slide Time: 06:06)



(Refer Slide Time: 06:17)



And then, if we just we have a 95 percent confidence band as well, ok.

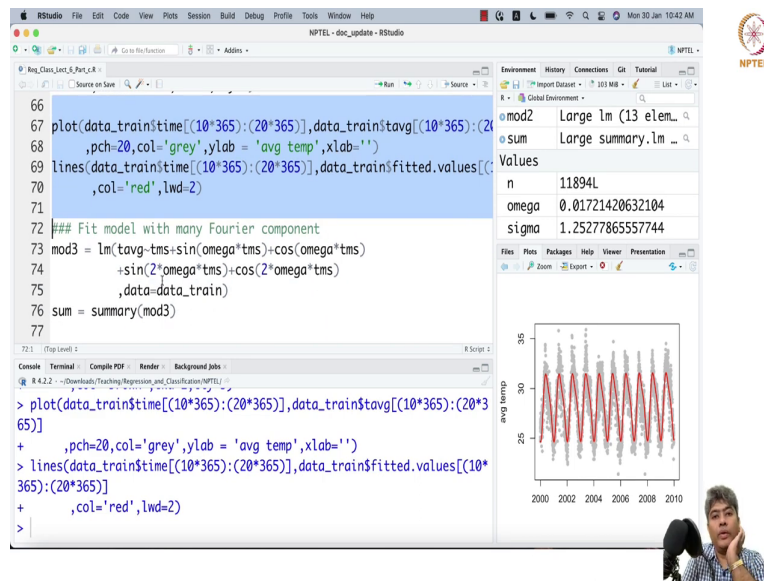
(Refer Slide Time: 06:29)

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for plotting training data and fitted values with confidence intervals. Lines 61-72 are highlighted in blue.
- Environment:** Shows the current environment with variables like 'mod2' and 'sum'.
- Values:** Displays the values of the fitted model parameters: n = 11894L, omega = 0.01721420632104, and sigma = 1.25277865557744.
- Console:** Shows the execution of the plotting commands, resulting in a plot of 'temp_avg' vs 'data_train\$time[(10 * 365)]'.
- Plot:** A line plot showing 'temp_avg' on the y-axis (ranging from 25 to 35) and 'data_train\$time[(10 * 365)]' on the x-axis (ranging from 1990 to 2000). The plot displays a periodic signal with red dots and a grey line.

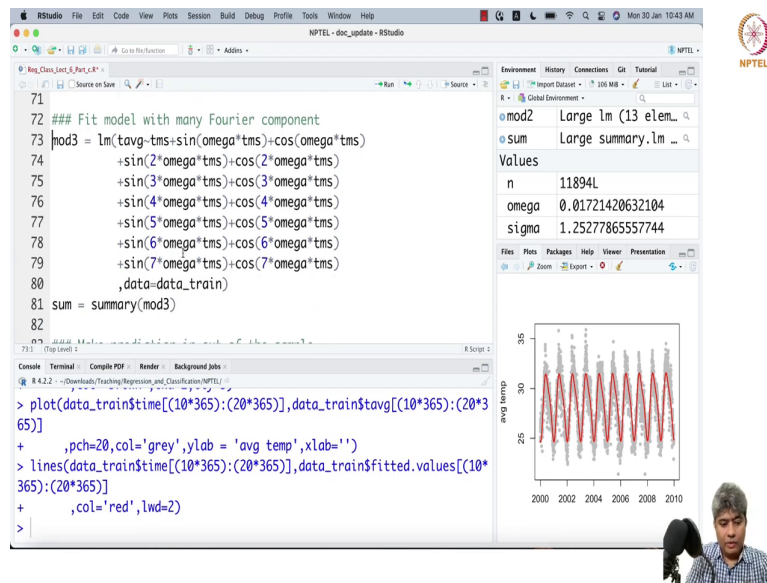
Now, and this is the from 2000 to 2010, ok.

(Refer Slide Time: 06:37)



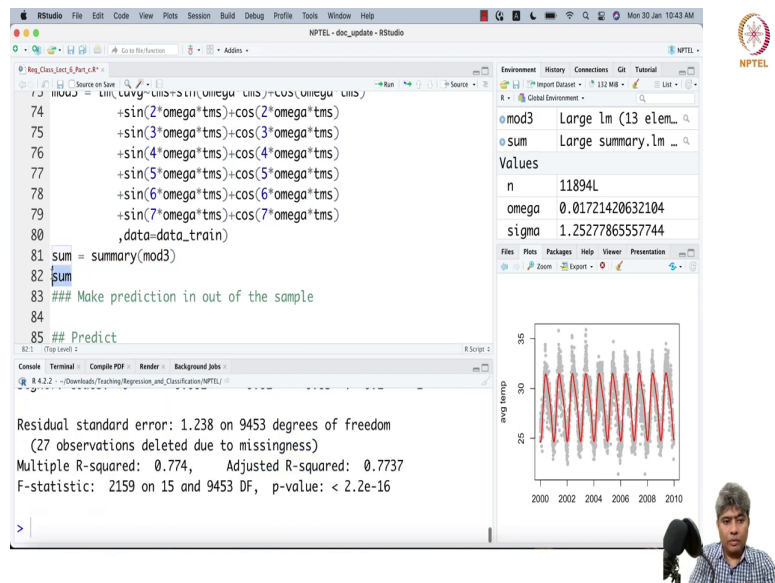
Now, we are going to add why we are only going to stop at 2 omega and cos 2 omega. We can in add as many cases as we want. Say may be up to 7 omega or 8 omega, ok 4, 4.

(Refer Slide Time: 06:47)



So, I am adding as many engineered feature as possible and I am not worried about over fitting much because I have too many datas, almost 11000 data points I have. So, I am not worried, ok. So, let me just try 7 Fourier terms.

(Refer Slide Time: 07:44)



The image shows a screenshot of the RStudio interface. The main editor window contains R code for fitting a linear model and making predictions. The console window displays the results of the model fit, including the residual standard error, R-squared values, and the F-statistic. A plot of the average temperature (avg temp) over time (year) is shown in the bottom right corner, with a red line representing the fitted model and grey points representing the data. A small inset image of a person is visible in the bottom right corner of the RStudio window.

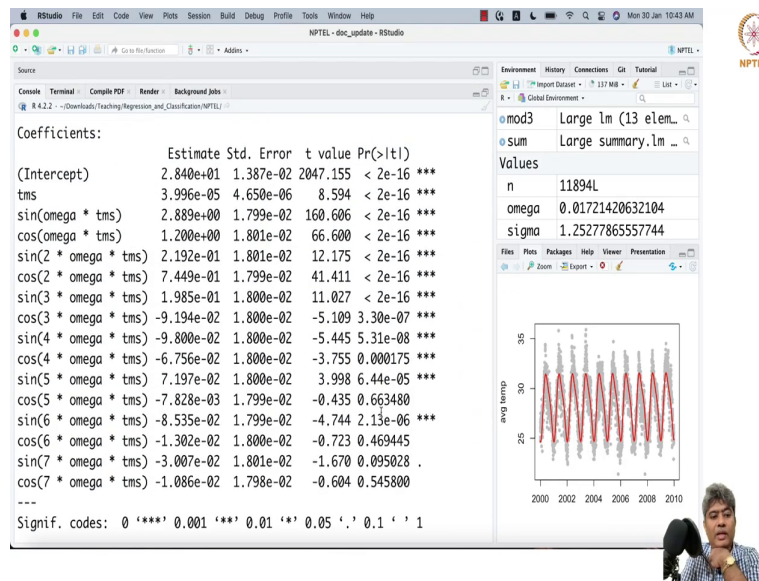
```
74 +sin(2*omega*tms)+cos(2*omega*tms)
75 +sin(3*omega*tms)+cos(3*omega*tms)
76 +sin(4*omega*tms)+cos(4*omega*tms)
77 +sin(5*omega*tms)+cos(5*omega*tms)
78 +sin(6*omega*tms)+cos(6*omega*tms)
79 +sin(7*omega*tms)+cos(7*omega*tms)
80 ,data=data_train)
81 sum = summary(mod3)
82 sum
83 ## Make prediction in out of the sample
84
85 ## Predict
```

Residual standard error: 1.238 on 9453 degrees of freedom
(27 observations deleted due to missingness)
Multiple R-squared: 0.774, Adjusted R-squared: 0.7737
F-statistic: 2159 on 15 and 9453 DF, p-value: < 2.2e-16

Parameter	Value
n	11894L
omega	0.01721420632104
sigma	1.2527786557744

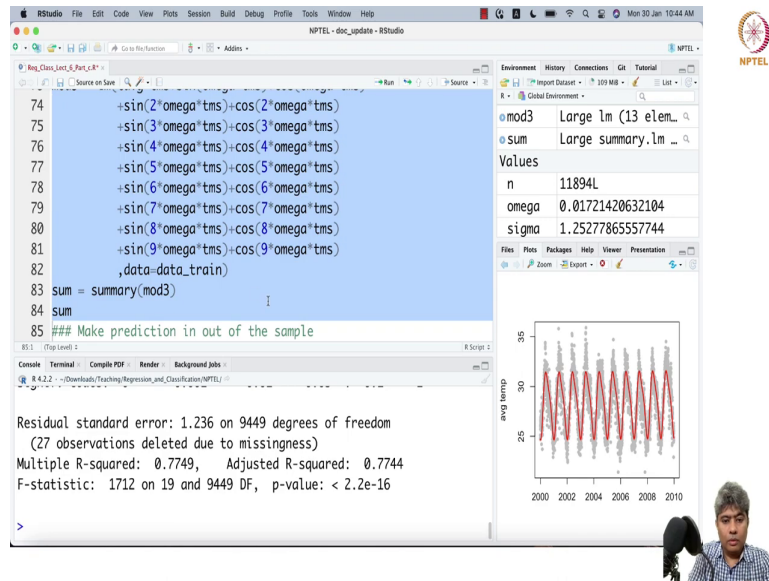
Plot: avg temp vs year (2000-2010). The plot shows a clear periodic pattern with a red line representing the fitted model and grey points representing the data.

(Refer Slide Time: 07:51)



And then, let us see, we can see that there are some Fourier terms like $\cos 5 \omega$ is not significant. Actually, in fact, after 3 ω onwards the significance has dropped to an extent though $\sin 6 \omega$ does have a effect, but $\cos 6 \omega$ 7ω they do not have a effect. You can even try few more, if you want know never know. If in case there are some higher things 9, 9. So, this is yeah.

(Refer Slide Time: 08:20)



The image shows a screenshot of the RStudio interface. The main editor window contains R code for fitting a linear model and summarizing it. The code is as follows:

```
74 +sin(2*omega*tms)+cos(2*omega*tms)
75 +sin(3*omega*tms)+cos(3*omega*tms)
76 +sin(4*omega*tms)+cos(4*omega*tms)
77 +sin(5*omega*tms)+cos(5*omega*tms)
78 +sin(6*omega*tms)+cos(6*omega*tms)
79 +sin(7*omega*tms)+cos(7*omega*tms)
80 +sin(8*omega*tms)+cos(8*omega*tms)
81 +sin(9*omega*tms)+cos(9*omega*tms)
82 ,data=data_train)
83 sum = summary(mod3)
84 sum
85 ### Make prediction in out of the sample
```

The console window displays the following output:

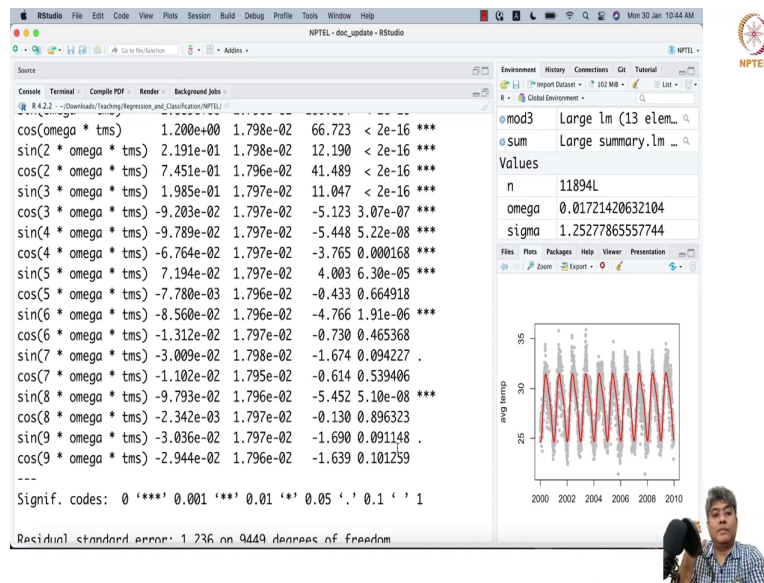
```
Residual standard error: 1.236 on 9449 degrees of freedom
(27 observations deleted due to missingness)
Multiple R-squared: 0.7749, Adjusted R-squared: 0.7744
F-statistic: 1712 on 19 and 9449 DF, p-value: < 2.2e-16
```

The Environment pane on the right shows the following values:

Variable	Value
n	11894L
omega	0.01721420632104
sigma	1.2527786557744

The plot window shows a time series plot of 'avg temp' from 2000 to 2010. The y-axis ranges from 25 to 35. The plot displays a clear periodic pattern with a red line representing the fitted model and grey points representing the data. A small inset image of a person is visible in the bottom right corner of the RStudio window.

(Refer Slide Time: 08:47)



So, yeah from 5, 6, 7, 8, may have, but 9; 8, 9 they do not have much effect. So, what we will do? We will is; we do not want to keep the keep growing the model size, ok. And if we do that we will see that there might be eventually over fitting which is happening.

(Refer Slide Time: 09:16)

The screenshot displays the RStudio interface. The main editor shows R code for fitting a linear model and making predictions. The console shows the output of the model fit, including coefficients and AIC values. The environment pane shows the fitted model objects. The plot pane shows a time series plot of average temperature.

```
R> # Fit a model with 9 variables
78 +sin(5*omega*tms)+cos(5*omega*tms)
79 +sin(6*omega*tms)+cos(6*omega*tms)
80 +sin(7*omega*tms)+cos(7*omega*tms)
81 +sin(8*omega*tms)+cos(8*omega*tms)
82 +sin(9*omega*tms)+cos(9*omega*tms)
83 sum = summary(mod3)
84 sum
85
86 mod4 = step(mod3)
87 ## Make prediction in out of the sample
88
89 ## Predict
R> predict(mod4, data=data_train)
```

Environment pane:

```
mod4 Large lm (14 elem...
sum Large summary.lm ...
```

Values:

```
n 11894L
omega 0.01721420632104
sigma 1.2527786557744
```

Console output:

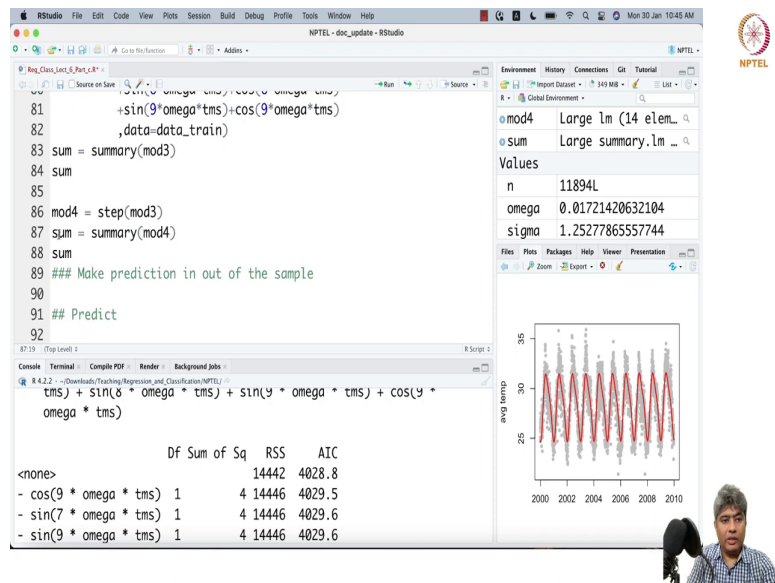
```
- tms 1 113 14555 4100.8
- sin(3 * omega * tms) 1 186 14628 4148.3
- sin(2 * omega * tms) 1 227 14669 4174.5
- cos(2 * omega * tms) 1 2630 17072 5611.2
- cos(omega * tms) 1 6803 21245 7681.8
- sin(omega * tms) 1 39555 53997 16514.6
```

Plot: avg temp vs time (2000-2010)

So, what we will do? We will keep this model, then the same time we will fit another model for model 4, model 4 and we will apply step wise variable selection on the model 3. So, if we just, you see it just fitted these models, so these models were fitted, ok. So, yeah here, from here.

So, this is the model that was fitted with this AIC, and then this is the model that was fitted, and eventually, it settled down to a model and let us see what was that.

(Refer Slide Time: 10:11)



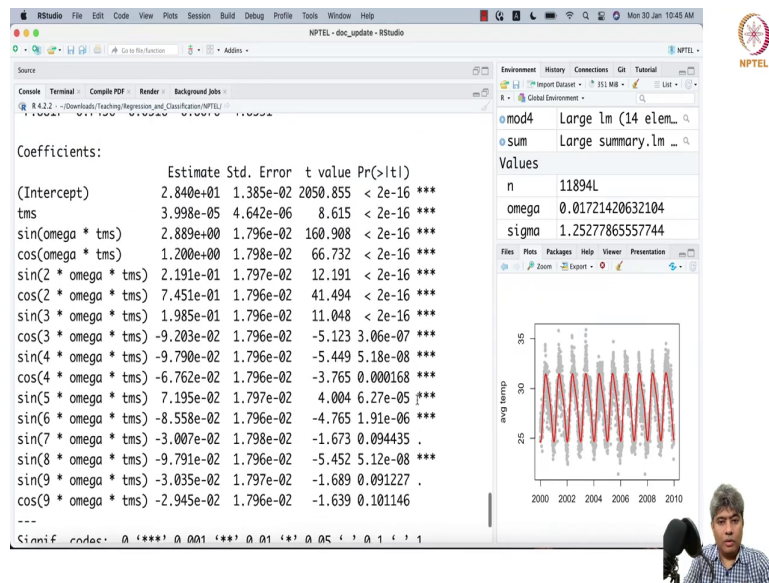
The image shows the RStudio interface with the following components:

- Source Editor:** Contains R code for fitting a linear model and making predictions.
- Environment:** Shows the fitted model objects: `mod4` (Large lm (14 elem...)) and `sum` (Large summary.lm ...).
- Values:** Displays the estimated parameters for the model: `n` (11894L), `omega` (0.01721420632104), and `sigma` (1.2527786557744).
- Console:** Shows the output of the model fit, including a table of coefficients and their corresponding Sum of Squares, Residual Sum of Squares (RSS), and Akaike Information Criterion (AIC).
- Plot:** A line plot titled "avg temp" showing the average temperature over time from 2000 to 2010. The y-axis ranges from 25 to 35. The plot shows a clear seasonal oscillation with a period of approximately 2 years.

```
81 +sin(9*omega*tms)+cos(9*omega*tms)
82 ,data=data_train)
83 sum = summary(mod3)
84 sum
85
86 mod4 = step(mod3)
87 sum = summary(mod4)
88 sum
89 ## Make prediction in out of the sample
90
91 ## Predict
92
```

	Df	Sum of Sq	RSS	AIC
<none>			14442	4028.8
- cos(9 * omega * tms)	1	4	14446	4029.5
- sin(7 * omega * tms)	1	4	14446	4029.6
- sin(9 * omega * tms)	1	4	14446	4029.6

(Refer Slide Time: 10:18)



The screenshot displays the RStudio interface. The console window shows the output of a linear model fit, including the coefficients table and a summary of the model. The plot window shows a time series plot of average temperature (avg temp) from 2000 to 2010, with a fitted model line and confidence intervals.

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.840e+01	1.385e-02	2050.855	< 2e-16 ***
tms	3.998e-05	4.642e-06	8.615	< 2e-16 ***
sin(omega * tms)	2.889e+00	1.796e-02	160.908	< 2e-16 ***
cos(omega * tms)	1.200e+00	1.798e-02	66.732	< 2e-16 ***
sin(2 * omega * tms)	2.191e-01	1.797e-02	12.191	< 2e-16 ***
cos(2 * omega * tms)	7.451e-01	1.796e-02	41.494	< 2e-16 ***
sin(3 * omega * tms)	1.985e-01	1.796e-02	11.048	< 2e-16 ***
cos(3 * omega * tms)	-9.203e-02	1.796e-02	-5.123	3.06e-07 ***
sin(4 * omega * tms)	-9.790e-02	1.796e-02	-5.449	5.18e-08 ***
cos(4 * omega * tms)	-6.762e-02	1.796e-02	-3.765	0.000168 ***
sin(5 * omega * tms)	7.195e-02	1.797e-02	4.004	6.27e-05 ***
sin(6 * omega * tms)	-8.558e-02	1.796e-02	-4.765	1.91e-06 ***
sin(7 * omega * tms)	-3.007e-02	1.798e-02	-1.673	0.094435 .
sin(8 * omega * tms)	-9.791e-02	1.796e-02	-5.452	5.12e-08 ***
sin(9 * omega * tms)	-3.035e-02	1.797e-02	-1.689	0.091227 .
cos(9 * omega * tms)	-2.945e-02	1.796e-02	-1.639	0.101146 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Values

n	11894L
omega	0.01721420632104
sigma	1.2527786557744

The plot shows 'avg temp' on the y-axis (ranging from 25 to 35) and years from 2000 to 2010 on the x-axis. The data points are represented by grey dots, and a red line shows the fitted model. The model exhibits a clear seasonal pattern with annual oscillations.

(Refer Slide Time: 10:24)

The image shows the RStudio interface with a linear model fit. The console displays the following coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.840e+01	1.385e-02	2050.855	< 2e-16 ***
tms	3.998e-05	4.642e-06	8.615	< 2e-16 ***
sin(omega * tms)	2.889e+00	1.796e-02	160.908	< 2e-16 ***
cos(omega * tms)	1.200e+00	1.798e-02	66.732	< 2e-16 ***
sin(2 * omega * tms)	2.191e-01	1.797e-02	12.191	< 2e-16 ***
cos(2 * omega * tms)	7.451e-01	1.796e-02	41.494	< 2e-16 ***
sin(3 * omega * tms)	1.985e-01	1.796e-02	11.048	< 2e-16 ***
cos(3 * omega * tms)	-9.203e-02	1.796e-02	-5.123	3.06e-07 ***
sin(4 * omega * tms)	-9.790e-02	1.796e-02	-5.449	5.18e-08 ***
cos(4 * omega * tms)	-6.762e-02	1.796e-02	-3.765	0.000168 ***
sin(5 * omega * tms)	7.195e-02	1.797e-02	4.004	6.27e-05 ***
sin(6 * omega * tms)	-8.558e-02	1.796e-02	-4.765	1.91e-06 ***
sin(7 * omega * tms)	-3.007e-02	1.798e-02	-1.673	0.094435 .
sin(8 * omega * tms)	-9.791e-02	1.796e-02	-5.452	5.12e-08 ***
sin(9 * omega * tms)	-3.035e-02	1.797e-02	-1.689	0.091227 .
cos(9 * omega * tms)	-2.945e-02	1.796e-02	-1.639	0.101146 .

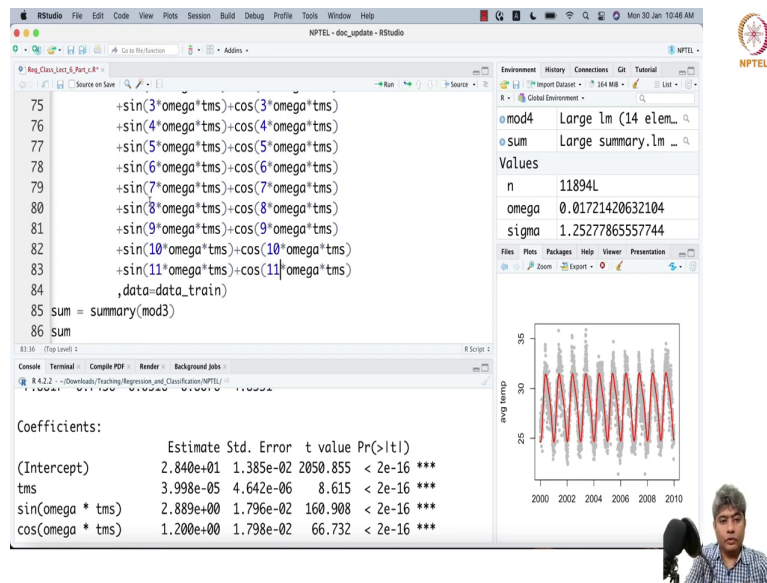
The plot shows 'avg hump' on the y-axis (ranging from 25 to 35) and time on the x-axis (ranging from 2000 to 2010). The plot displays a series of red vertical lines representing individual observations and a black line representing the fitted model. The fitted model shows a clear periodic pattern with a mean value around 30.

The RStudio interface also shows the Environment pane with the following values:

Variable	Value
n	11894L
omega	0.01721420632104
sigma	1.2527786557744

So, I can just copy and paste it here, and I taking some summary. So, it has drop the cos terms and keep only the sine terms that has effect in the model though it is it did not throw away the cos 9.

(Refer Slide Time: 10:41)



So, we can actually; that means, we can add few more terms.

(Refer Slide Time: 11:00)

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

```
77 +sin(4*omega*tms)+cos(4*omega*tms)
78 +sin(5*omega*tms)+cos(5*omega*tms)
79 +sin(6*omega*tms)+cos(6*omega*tms)
80 +sin(7*omega*tms)+cos(7*omega*tms)
81 +sin(8*omega*tms)+cos(8*omega*tms)
82 +sin(9*omega*tms)+cos(9*omega*tms)
83 +sin(10*omega*tms)+cos(10*omega*tms)
84 +sin(11*omega*tms)+cos(11*omega*tms)
85 ,data=data_train)
86 sum = summary(mod3)
87 sum
88 mod4 = step(mod3)
```

The Environment pane on the right shows the following objects:

- mod3: Large lm (13 elem...)
- mod4: Large lm (14 elem...)
- sum: Large summary.lm ...

The Values pane shows the following values:

- n: 11894L
- omega: 0.01721420632104

The Console window shows the output of the code:

```
+ +sin(7*omega*tms)+cos(7*omega*tms)
+ +sin(8*omega*tms)+cos(8*omega*tms)
+ +sin(9*omega*tms)+cos(9*omega*tms)
+ +sin(10*omega*tms)+cos(10*omega*tms)
+ +sin(11*omega*tms)+cos(11*omega*tms)
+ ,data=data_train
>
```

The Plot pane shows a line graph of the average response (avg resp) over time (tms). The x-axis ranges from 2000 to 2010, and the y-axis ranges from 25 to 35. The plot shows a periodic oscillation with a period of approximately 2 years. The NPTEL logo is visible in the top right corner of the RStudio window.

(Refer Slide Time: 11:09)

The image shows the RStudio interface with the following components:

- Source Editor:** Contains R code for fitting linear models and summarizing them. Lines 78-84 define a data frame with columns for sine and cosine terms of order 6 to 11. Lines 85-89 fit a model 'mod3' and summarize it, then fit a model 'mod4' and summarize it.
- Environment:** Shows objects 'mod3', 'mod4', and 'sum'. The 'Values' section shows 'n' as 11894L and 'omega' as 0.01721420632104.
- Console:** Displays the output of the fit.summary() function for 'mod3', showing coefficients for sin(7*omega*tms) through sin(10*omega*tms) and their corresponding p-values.
- Plot:** A line plot titled 'avg temp' showing temperature fluctuations from 2000 to 2010. The y-axis ranges from 25 to 35. The plot shows a clear periodic pattern with red data points and a fitted model line.

Console Output:

```
sin(7 * omega * tms) -3.014e-02 1.797e-02 -1.677 0.093578 .
cos(7 * omega * tms) -1.099e-02 1.794e-02 -0.613 0.540178
sin(8 * omega * tms) -9.798e-02 1.795e-02 -5.457 4.96e-08 ***
cos(8 * omega * tms) -2.283e-03 1.796e-02 -0.127 0.898880
sin(9 * omega * tms) -3.032e-02 1.796e-02 -1.688 0.091405 .
cos(9 * omega * tms) -2.935e-02 1.796e-02 -1.635 0.102146
sin(10 * omega * tms) 1.676e-02 1.795e-02 0.934 0.350520
cos(10 * omega * tms) 5.571e-03 1.795e-02 0.210 0.831025 ***
```



(Refer Slide Time: 11:10)

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for fitting a linear model and making predictions. The code includes:

```
87 mod4 = step(mod3)
88 sum = summary(mod4)
89 sum
90
91 ## Make prediction in out of the sample
92
93 ## Predict
94
95 data_test$pred = NA
96 data_test$pred = predict(mod1,newdata = data_test )
97
98 ## Out-sample accuracy
```
- Environment Pane:** Shows three objects: 'mod3' (Large lm (13 elem...)), 'mod4' (Large lm (14 elem...)), and 'sum' (Large summary.lm...). Below this, the 'Values' section shows:

```
n      11894L
omega  0.01721420632104
```
- Console:** Displays the output of trigonometric functions:

```
sin(7 * omega * tms) -3.014e-02  1.797e-02 -1.677  0.093578 .
cos(7 * omega * tms) -1.099e-02  1.794e-02 -0.613  0.540178
sin(8 * omega * tms) -9.798e-02  1.795e-02 -5.457  4.96e-08 ***
cos(8 * omega * tms) -2.283e-03  1.796e-02 -0.127  0.898880
sin(9 * omega * tms) -3.032e-02  1.796e-02 -1.688  0.091405 .
cos(9 * omega * tms) -2.935e-02  1.796e-02 -1.635  0.102146
sin(10 * omega * tms)  1.676e-02  1.795e-02  0.934  0.350520
cos(10 * omega * tms)  5.571e-02  1.795e-02  2.101  0.001025 ***
```
- Plot:** A line plot titled 'avg temp' showing temperature fluctuations from 2000 to 2010. The y-axis ranges from 25 to 35. The plot shows a clear periodic pattern with red data points and a fitted model line.

And let me try this and so yeah looks good, right.

(Refer Slide Time: 11:16)

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains R code for fitting a model, summarizing it, and making predictions on new data.
- Environment:** Lists objects 'mod3', 'mod4', and 'sum' as 'Large lm' or 'Large summary.lm' objects.
- Values:** Shows the values for 'n' (11894L) and 'omega' (0.01721420632104).
- Console:** Displays the output of the `summary(mod4)` command, showing coefficients for various trigonometric functions of time.
- Plot:** A line plot titled 'avg temp' showing a periodic oscillation of temperature over time from 2000 to 2010.

```
87
88 mod4 = step(mod3)
89 sum = summary(mod4)
90 sum
91 ## Make prediction in out of the sample
92
93 ## Predict
94
95 data_test$pred = NA
96 data_test$pred = predict(mod1,newdata = data_test )
97
98 ## Out-sample accuracy
```

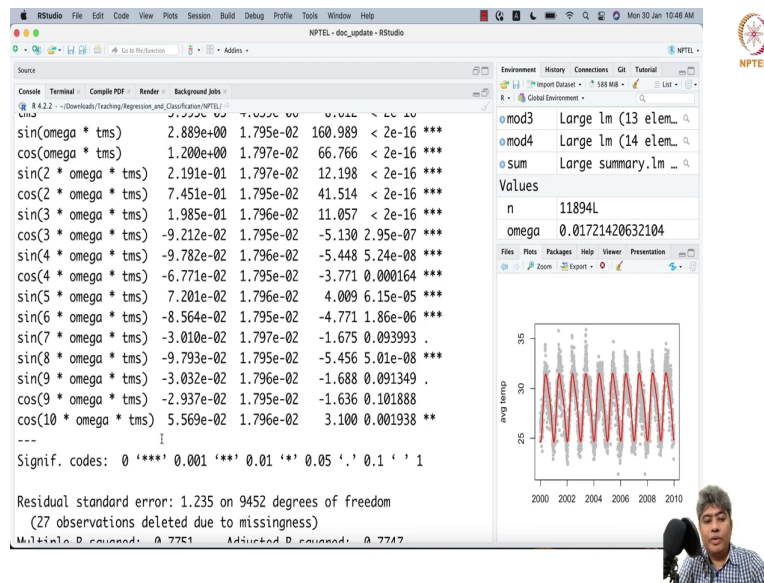
- tms	1	113	14540	4093.2
- sin(3 * omega * tms)	1	187	14614	4140.8
- sin(2 * omega * tms)	1	227	14654	4167.1
- cos(2 * omega * tms)	1	2630	17057	5605.1
- cos(omega * tms)	1	6804	21231	7677.6
- sin(omega * tms)	1	39559	53986	16514.7

avg temp

2000 2002 2004 2006 2008 2010

And then let me just run this stepwise forward section, and see just run and let me just write.

(Refer Slide Time: 11:21)

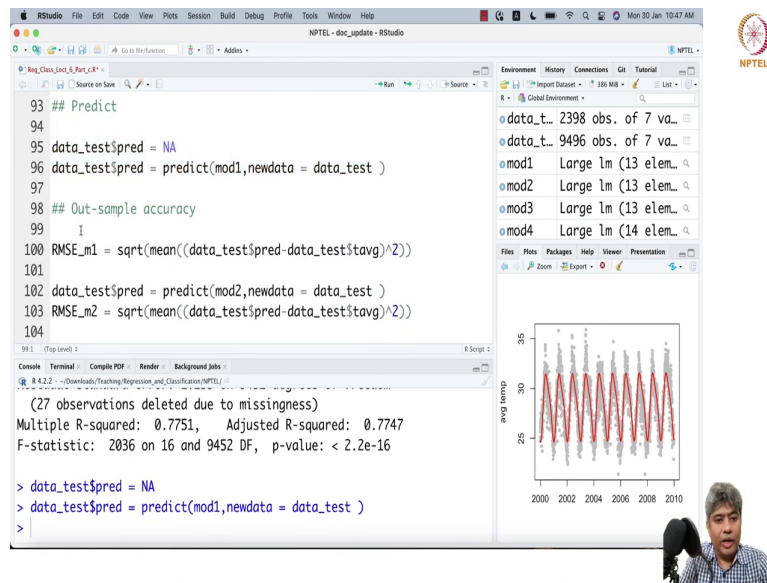


The screenshot shows the RStudio interface with the following components:

- Console:** Displays the output of a linear regression model. The coefficients for terms $\sin(\omega * tms)$ through $\cos(10 * \omega * tms)$ are shown with their standard errors, t-statistics, and p-values. The p-values for the 11th and 12th terms are significantly higher than the others, indicating they are not statistically significant. The output ends with: "Residual standard error: 1.235 on 9452 degrees of freedom (27 observations deleted due to missingness) Multiple R-squared: 0.7751 Adjusted R-squared: 0.7747".
- Environment:** Shows the model objects: mod3, mod4, and sum.
- Values:** Shows the values for the parameters: n = 11894L and omega = 0.01721420632104.
- Plot:** A line plot titled "avg temp" showing the average temperature over time from 2000 to 2010. The y-axis ranges from 25 to 35. The plot shows a clear seasonal oscillation with a period of approximately 2 years.

So, now it has completely drop the 11th Fourier term and it stopped at 10th Fourier term. So, we can just stop at here. So, we can just. Now, what we will do? We will do the prediction. So, first we will do prediction and we will calculate the out of the sample accuracy.

(Refer Slide Time: 11:46)



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

```
93 ## Predict
94
95 data_test$pred = NA
96 data_test$pred = predict(mod1,newdata = data_test )
97
98 ## Out-sample accuracy
99 I
100 RMSE_m1 = sqrt(mean((data_test$pred-data_test$avg)^2))
101
102 data_test$pred = predict(mod2,newdata = data_test )
103 RMSE_m2 = sqrt(mean((data_test$pred-data_test$avg)^2))
104
```

The console window shows the following output:

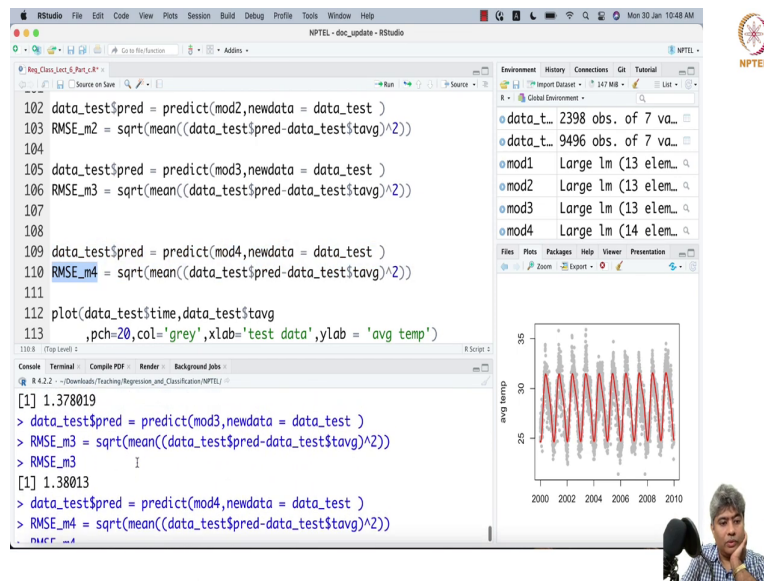
```
(27 observations deleted due to missingness)
Multiple R-squared: 0.7751, Adjusted R-squared: 0.7747
F-statistic: 2036 on 16 and 9452 DF, p-value: < 2.2e-16

> data_test$pred = NA
> data_test$pred = predict(mod1,newdata = data_test )
>
```

The Environment pane on the right lists several objects: data_t_ (2398 obs. of 7 va...), data_t_ (9496 obs. of 7 va...), mod1 (Large lm (13 elem...)), mod2 (Large lm (13 elem...)), mod3 (Large lm (13 elem...)), and mod4 (Large lm (14 elem...)). Below the Environment pane is a plot of 'avg temp' over time (2000-2010), showing a clear seasonal pattern with peaks around 30 and troughs around 25.

So, in the test, we are creating prediction and here is the out of the sample accuracy and then we will do. So, what was the RMSE? So, 1.5 c for the first model. Then, if we do prediction for the second model and I may see would be 1.37.

(Refer Slide Time: 12:15)



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

```
102 data_test$pred = predict(mod2,newdata = data_test )
103 RMSE_m2 = sqrt(mean((data_test$pred-data_test$avg)^2))
104
105 data_test$pred = predict(mod3,newdata = data_test )
106 RMSE_m3 = sqrt(mean((data_test$pred-data_test$avg)^2))
107
108
109 data_test$pred = predict(mod4,newdata = data_test )
110 RMSE_m4 = sqrt(mean((data_test$pred-data_test$avg)^2))
111
112 plot(data_test$time,data_test$avg
113      ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')
```

The console window shows the following output:

```
[1] 1.378019
> data_test$pred = predict(mod3,newdata = data_test )
> RMSE_m3 = sqrt(mean((data_test$pred-data_test$avg)^2))
> RMSE_m3
[1] 1.38013
> data_test$pred = predict(mod4,newdata = data_test )
> RMSE_m4 = sqrt(mean((data_test$pred-data_test$avg)^2))
[1] 1.38013
```

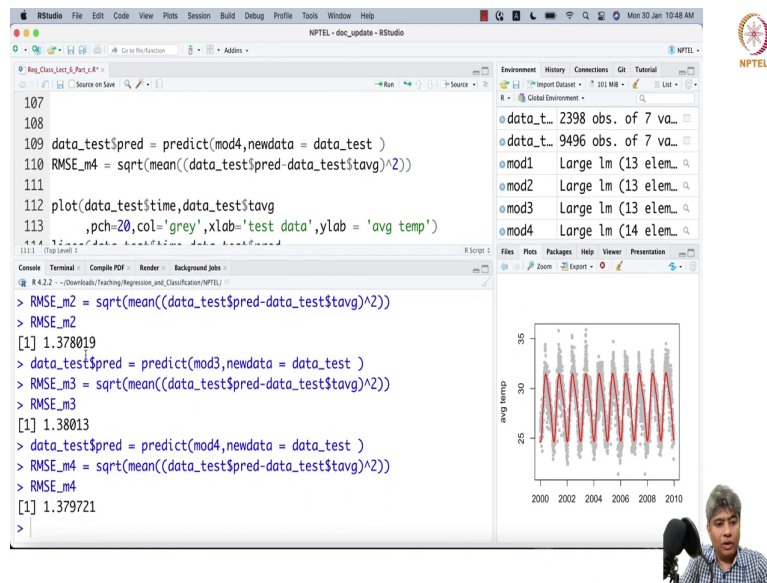
The Environment pane on the right lists several objects: data_t_ (2398 obs. of 7 va...), data_t_ (9496 obs. of 7 va...), mod1 (Large lm (13 elem...), mod2 (Large lm (13 elem...), mod3 (Large lm (13 elem...), and mod4 (Large lm (14 elem...)).

The plot window shows a time series plot of average temperature (avg temp) over time (test data). The x-axis ranges from 2000 to 2010, and the y-axis ranges from 25 to 35. The plot displays a regular oscillating pattern with grey points and a red line representing the model fit.

Now, if we do the same thing for the third model, let us done that and the RMSE 3 is 1.38. Now, you can see that RMSE has gone up because we have added too many models. So, that we can see too many Fourier term, too many engineered term. And as a result, you can see that there are some over fitting tendencies being picked up, and then, we have the step wise selected, for selective model.

So, from there if we calculate the RMSE out of the sample RMSE is 1.379. So, this is where it is stopping.

(Refer Slide Time: 13:18)



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

```
107
108
109 data_test$pred = predict(mod4,newdata = data_test )
110 RMSE_m4 = sqrt(mean((data_test$pred-data_test$avg)^2))
111
112 plot(data_test$time,data_test$avg
113       ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')
```

The console window shows the execution of the code and the resulting RMSE values for four different models:

```
> RMSE_m2 = sqrt(mean((data_test$pred-data_test$avg)^2))
> RMSE_m2
[1] 1.378019
> data_test$pred = predict(mod3,newdata = data_test )
> RMSE_m3 = sqrt(mean((data_test$pred-data_test$avg)^2))
> RMSE_m3
[1] 1.38013
> data_test$pred = predict(mod4,newdata = data_test )
> RMSE_m4 = sqrt(mean((data_test$pred-data_test$avg)^2))
> RMSE_m4
[1] 1.379721
>
```

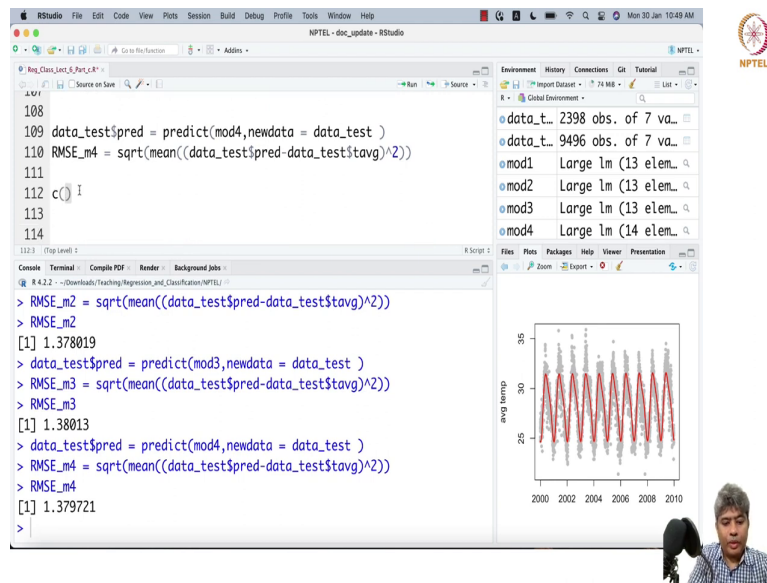
The Environment pane on the right lists several objects: data_t... (2398 obs. of 7 va...), data_t... (9496 obs. of 7 va...), mod1 (Large lm (13 elem...), mod2 (Large lm (13 elem...), mod3 (Large lm (13 elem...), and mod4 (Large lm (14 elem...)).

The plot window shows a time series plot of average temperature (avg temp) over time (test data). The x-axis ranges from 2000 to 2010, and the y-axis ranges from 25 to 35. The plot displays a clear seasonal pattern with peaks around 30 and troughs around 25. The data points are represented by grey circles, and a red line represents the fitted model.

The NPTEL logo is visible in the top right corner of the RStudio window.

So, either we can go for 4th model or we can go for because end of the day the simpler 1.378. The simpler say second model with 2 sine cosine Fourier transform has a lower RMSE than a bigger more complex model, even if after doing a stepwise selection.

(Refer Slide Time: 13:48)



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

```
108  
109 data_test$pred = predict(mod4,newdata = data_test )  
110 RMSE_m4 = sqrt(mean((data_test$pred-data_test$avg)^2))  
111  
112 c() I  
113  
114
```

The console window shows the execution of the code and the resulting RMSE values for four models:

```
> RMSE_m2 = sqrt(mean((data_test$pred-data_test$avg)^2))  
> RMSE_m2  
[1] 1.378019  
> data_test$pred = predict(mod3,newdata = data_test )  
> RMSE_m3 = sqrt(mean((data_test$pred-data_test$avg)^2))  
> RMSE_m3  
[1] 1.38013  
> data_test$pred = predict(mod4,newdata = data_test )  
> RMSE_m4 = sqrt(mean((data_test$pred-data_test$avg)^2))  
> RMSE_m4  
[1] 1.379721  
>
```

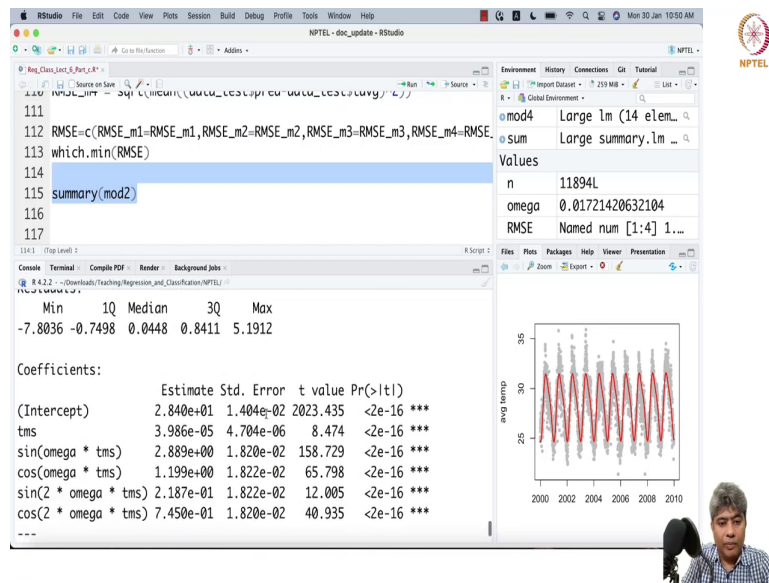
The Environment pane on the right lists several objects: data_t... (2398 obs. of 7 va...), data_t... (9496 obs. of 7 va...), mod1 (Large lm (13 elem...), mod2 (Large lm (13 elem...), mod3 (Large lm (13 elem...), and mod4 (Large lm (14 elem...)).

A plot titled 'avg temp' is visible in the bottom right corner, showing a time series of average temperature from 2000 to 2010. The y-axis ranges from 25 to 35. The plot shows a clear seasonal pattern with peaks around 30-32 and troughs around 25-27.

The NPTEL logo is visible in the top right corner of the RStudio window.

So, if I have to choose out of these 3, I will go for the second model. So, if I have to let me just you know let me just is equal to RMSE 1, RMSE 2 equal to RMSE 2, RMSE 3 equal to RMSE 3, RMSE 4 equal to RMSE 4, ok. So, clearly, if I just say RMSE is this and which dot min of RMSE; clearly the second one is the RMSE m2, second model has the maximum minimum RMSE.

(Refer Slide Time: 13:48)



The screenshot shows the RStudio interface with the following content:

```
111 RMSE=c(RMSE_m1=RMSE_m1, RMSE_m2=RMSE_m2, RMSE_m3=RMSE_m3, RMSE_m4=RMSE_m4)
112 which.min(RMSE)
113
114
115 summary(mod2)
116
117
```

Environment: Large lm (14 elements), Large summary.lm

Values:

n	11894L
omega	0.01721420632104
RMSE	Named num [1:4] 1...

Console (Top Level):

```
Min 1Q Median 3Q Max
-7.8036 -0.7498 0.0448 0.8411 5.1912
```

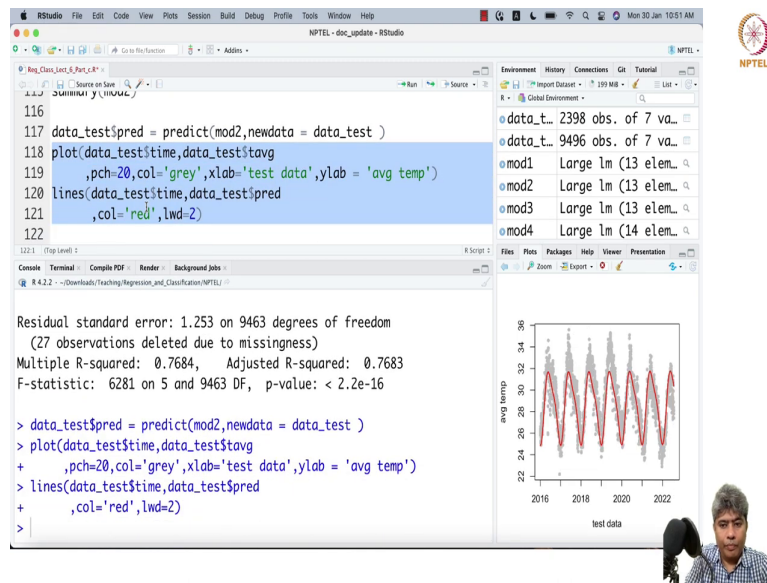
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.840e+01	1.404e-02	2023.435	<2e-16 ***
tms	3.986e-05	4.704e-06	8.474	<2e-16 ***
sin(omega * tms)	2.889e+00	1.820e-02	158.729	<2e-16 ***
cos(omega * tms)	1.199e+00	1.822e-02	65.798	<2e-16 ***
sin(2 * omega * tms)	2.187e-01	1.822e-02	12.005	<2e-16 ***
cos(2 * omega * tms)	7.450e-01	1.820e-02	40.935	<2e-16 ***

Plot: A line graph showing 'avg amp' on the y-axis (ranging from 25 to 35) and time on the x-axis (ranging from 2000 to 2010). The plot displays a highly oscillatory signal with a red line and grey dots, indicating significant overfitting.

So, even if after doing the statewise selection, significant dimension reduction, the model is still complex and doing some bit of over fitting perhaps. So, we can do summary of mod 2.

(Refer Slide Time: 15:33)



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

```
116  
117 data_test$pred = predict(mod2,newdata = data_test )  
118 plot(data_test$time,data_test$avg  
119       ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')  
120 lines(data_test$time,data_test$pred  
121       ,col='red',lwd=2)  
122
```

The console window shows the output of the code execution:

```
Residual standard error: 1.253 on 9463 degrees of freedom  
(27 observations deleted due to missingness)  
Multiple R-squared: 0.7684, Adjusted R-squared: 0.7683  
F-statistic: 6281 on 5 and 9463 DF, p-value: < 2.2e-16  
  
> data_test$pred = predict(mod2,newdata = data_test )  
> plot(data_test$time,data_test$avg  
+       ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')  
> lines(data_test$time,data_test$pred  
+       ,col='red',lwd=2)  
>
```

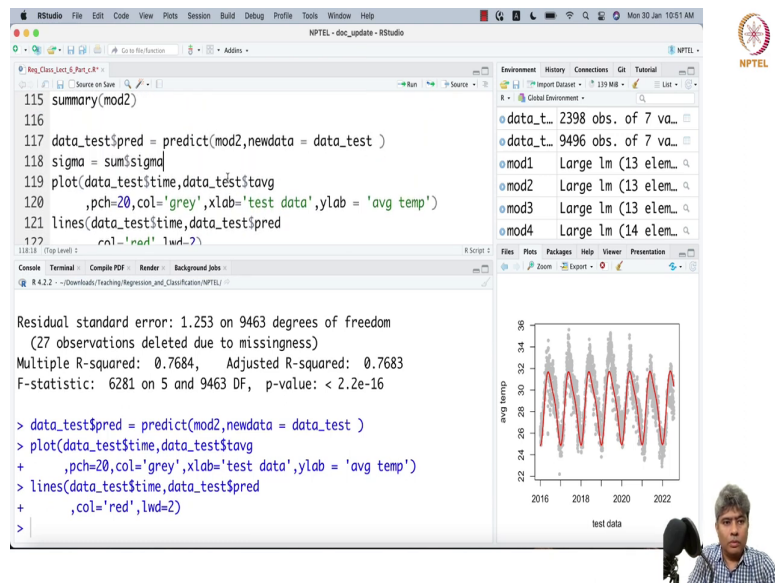
The Environment pane on the right lists several objects: data_t_ (2398 obs. of 7 va...), data_t_ (9496 obs. of 7 va...), mod1 (Large lm (13 elem...)), mod2 (Large lm (13 elem...)), mod3 (Large lm (13 elem...)), and mod4 (Large lm (14 elem...)).

The plot window shows a time series plot of average temperature (avg temp) over time (test data). The x-axis ranges from 2016 to 2022, and the y-axis ranges from 22 to 36. The plot displays a regular oscillating pattern with grey points and a red line representing the predicted values.

The NPTEL logo is visible in the top right corner of the RStudio window.

So, we can look into the parsimonious model. This is our parsimonious model. Perhaps, we just be happy with this. We just take this line and run once more. With this, we want this and sigma, only the sigma of this, correct sigma for this model. We have to just go up.

(Refer Slide Time: 16:09)



The image displays the RStudio interface with the following components:

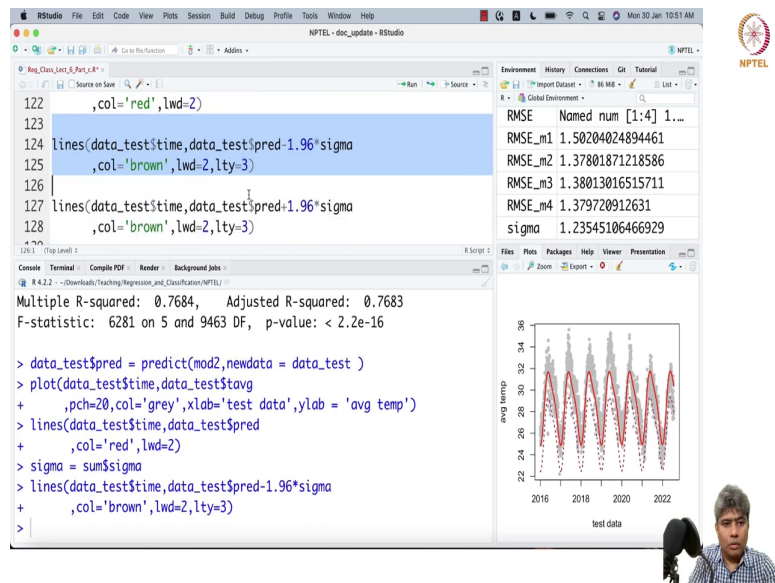
- Source Editor:** Contains R code for model prediction and plotting:

```
115 summary(mod2)
116
117 data_test$pred = predict(mod2,newdata = data_test )
118 sigma = sum(sigma)
119 plot(data_test$time,data_test$avg
120       ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')
121 lines(data_test$time,data_test$pred
122       ,col='red',lwd=2)
123
```
- Environment:** Lists objects in the workspace:
 - data_t... 2398 obs. of 7 va...
 - data_t... 9496 obs. of 7 va...
 - mod1 Large lm (13 elem...
 - mod2 Large lm (13 elem...
 - mod3 Large lm (13 elem...
 - mod4 Large lm (14 elem...
- Console:** Shows the output of the code execution:

```
Residual standard error: 1.253 on 9463 degrees of freedom
(27 observations deleted due to missingness)
Multiple R-squared: 0.7684, Adjusted R-squared: 0.7683
F-statistic: 6281 on 5 and 9463 DF, p-value: < 2.2e-16

> data_test$pred = predict(mod2,newdata = data_test )
> plot(data_test$time,data_test$avg
+       ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')
> lines(data_test$time,data_test$pred
+       ,col='red',lwd=2)
>
```
- Plot:** A scatter plot of 'avg temp' vs 'test data' from 2016 to 2022. The y-axis ranges from 22 to 36. Grey points represent observed data, and a red line represents the model's prediction. The plot shows a clear seasonal pattern.
- NPTEL Logo:** Located in the top right corner.
- Speaker:** A small inset image of a person's head and shoulders is visible in the bottom right corner.

(Refer Slide Time: 16:19)



The image displays the RStudio interface with the following components:

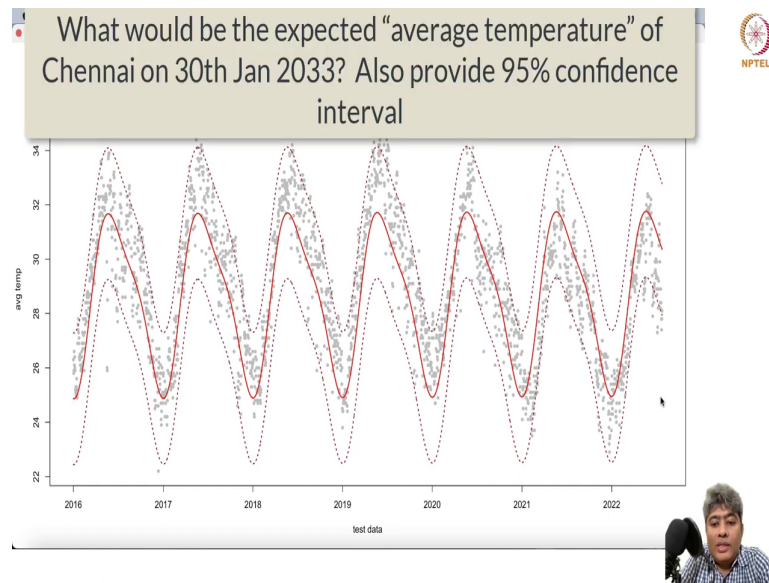
- Code Editor:** Lines 122-128 show R code for plotting data and prediction intervals. Line 124 is highlighted in blue.
- Environment:** Shows variables like RMSE_m1 through RMSE_m4 and sigma.
- Console:** Displays model statistics: Multiple R-squared: 0.7684, Adjusted R-squared: 0.7683, F-statistic: 6281 on 5 and 9463 DF, p-value: < 2.2e-16. Below this is the R code for the plot.
- Plot:** A line plot titled 'test data' showing 'avg temp' on the y-axis (ranging from 22 to 36) and 'test data' on the x-axis (years 2016 to 2022). The plot includes grey dots for observed data, a red line for the predicted mean, and brown shaded areas for the 1.96-sigma prediction intervals.

```
122 ,col='red',lwd=2)
123
124 lines(data_test$time,data_test$pred-1.96*sigma
125 ,col='brown',lwd=2,lty=3)
126
127 lines(data_test$time,data_test$pred+1.96*sigma
128 ,col='brown',lwd=2,lty=3)

Multiple R-squared: 0.7684, Adjusted R-squared: 0.7683
F-statistic: 6281 on 5 and 9463 DF, p-value: < 2.2e-16

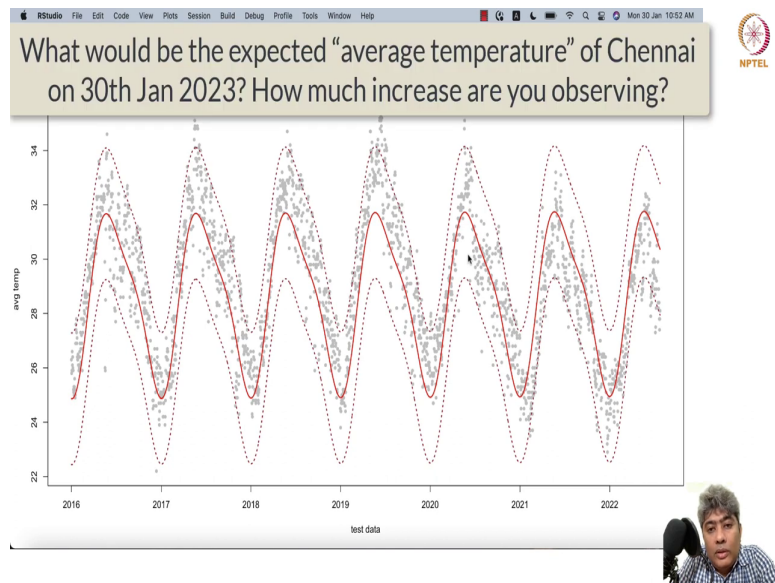
> data_test$pred = predict(mod2,newdata = data_test )
> plot(data_test$time,data_test$avg
+ ,pch=20,col='grey',xlab='test data',ylab = 'avg temp')
> lines(data_test$time,data_test$pred
+ ,col='red',lwd=2)
> sigma = sum$sigma
> lines(data_test$time,data_test$pred-1.96*sigma
+ ,col='brown',lwd=2,lty=3)
>
```

(Refer Slide Time: 16:26)



So, now if we just look into this, so this is our final out of the sample performance from 2016 to 2022. Now, here is one self-assessed assignment for you guys use this model and can you find what would be the temperature of Chennai, average expected temperature of Chennai in January in year 2033, that is 10 years from today. In January, say 30th January, what would be the expected temperature? Can you with this kind of confidence interval can you figure that out?

(Refer Slide Time: 17:25)



And if, what is the expected temperature and in the 30th January on 2023, and how much increase you are observing. Can you do this self assessed exercise and let me know that what is your findings? That will be really fun exercise. And I think that will be we all want to know that, the answer to that question.

Thank you very much. See you in the next lecture. Bye.