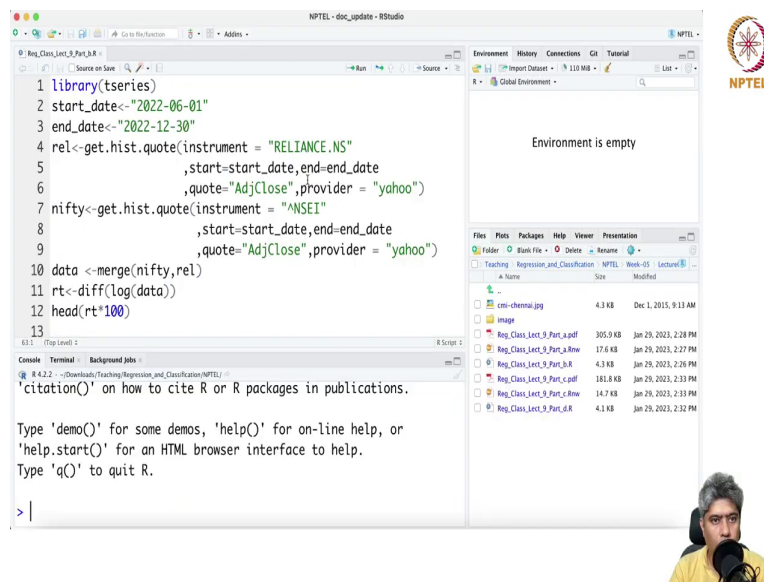


Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Chennai Mathematical Institute

Lecture - 31
Hands on with R for CAPM

Hello all welcome back to the part B of lecture 9. In this video lecture, we will do some hands on with R. So, let me first open my R, RStudio.

(Refer Slide Time: 00:34)





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

```
1 library(tseries)
2 start_date<-"2022-06-01"
3 end_date<-"2022-12-30"
4 rel<-get.hist.quote(instrument = "RELIANCE.NS"
5                     ,start=start_date,end=end_date
6                     ,quote="AdjClose",provider = "yahoo")
7 nifty<-get.hist.quote(instrument = "NSEI"
8                       ,start=start_date,end=end_date
9                       ,quote="AdjClose",provider = "yahoo")
10 data <-merge(nifty,rel)
11 rt<-diff(log(data))
12 head(rt*100)
13
```

The console output shows the following text:

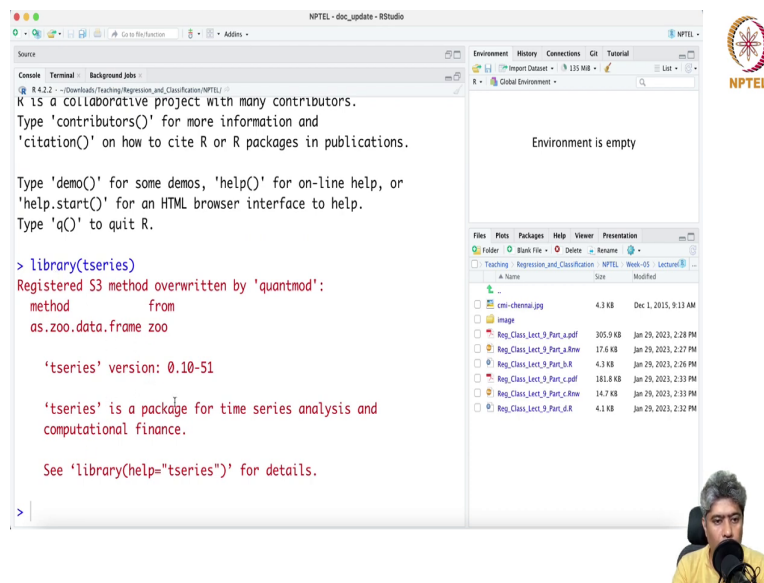
```
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> |
```

The Environment pane on the right shows "Environment is empty". The Files pane on the bottom right shows a list of files in the current directory.



So, this is the code that I have partially showed you in the, in my part first part of this lecture 9, where I am going to use this first line, the live T series package. In case you do not have T series package in your. So, if you run this.

(Refer Slide Time: 01:01)



The screenshot shows the RStudio interface with the following content:

```
Source
Console Terminal Background jobs
R 4.2.2 -> [Downloads/Teaching/Regression_and_Classification/NPTEL] >
K 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.

> library(tseries)
Registered S3 method overwritten by 'quantmod':
  method      from
as.zoo.data.frame zoo

'tseries' version: 0.10-51

'tseries' is a package for time series analysis and
computational finance.


See 'library(help="tseries")' for details.

>
```

Environment: Environment is empty

Files:

Name	Size	Modified
..		
cmi-chemical.jpg	4.3 KB	Dec 1, 2015, 9:13 AM
image		
Reg_Class_Lect_9_Part_a.pdf	305.9 KB	Jan 20, 2023, 2:28 PM
Reg_Class_Lect_9_Part_a.Rnw	17.6 KB	Jan 20, 2023, 2:27 PM
Reg_Class_Lect_9_Part_b.R	4.3 KB	Jan 20, 2023, 2:26 PM
Reg_Class_Lect_9_Part_c.pdf	181.8 KB	Jan 20, 2023, 2:33 PM
Reg_Class_Lect_9_Part_c.Rnw	14.7 KB	Jan 20, 2023, 2:33 PM
Reg_Class_Lect_9_Part_d.R	4.1 KB	Jan 20, 2023, 2:32 PM



So, in this case you may get a like you know it is this kind of information. If you do not have T series package then in your R install, then what you can do you can see here if there is a package you can see ok.

(Refer Slide Time: 01:21)

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

```
1 library(tseries)
2 start_date<- "2022-06-01"
3 end_date<- "2022-12-30"
4 rel<-get.hist.quote(instrument = "RELIANCE.NS"
5 ,start=start_date,end=end_date
6 ,quote="AdjClose",provider = "yahoo")
7 nifty<-get.hist.quote(instrument
8 ,start=star
9 ,quote="Adj
10 data <-merge(nifty,rel)
11 rt<-diff(log(data))
12 head(rt*100)
13
```

An "Install Packages" dialog box is open, showing the "Install from:" field set to "Repository (CRAN)". The "Install to Library:" field is set to the user's library path. The "Install dependencies" checkbox is checked. The "Install" button is highlighted.

The console output shows the following message:

```
'tseries' is a package for time series analysis and
computational finance.
See 'library(help="tseries")' for details.
```

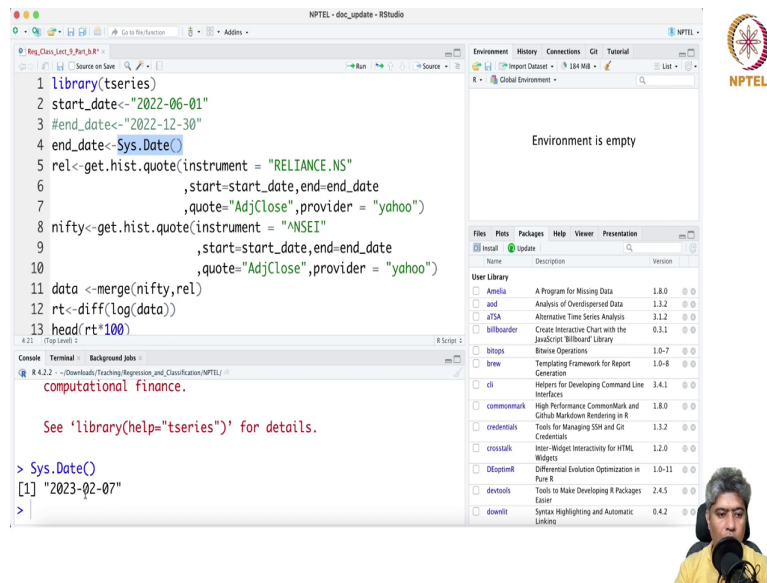
The "Environment" pane on the right shows "Environment is empty". The "Packages" pane on the right shows a list of installed and available packages, including 'tseries'.

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

In package you can click on there you can go to install and then if you write tseries and make sure your install dependency is corrected check and then you click install. Now, I am not going to install because it is already installed in my R environment. So, I am cancelling it.

Now, I am going to consider the start date as 1st June 2022 and in the analysis that I have shown in the package in the previous video I have considered 30th December 2022.

(Refer Slide Time: 02:11)



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

```
1 library(tseries)
2 start_date<- "2022-06-01"
3 #end_date<- "2022-12-30"
4 end_date<- Sys.Date()
5 rel<-get.hist.quote(instrument = "RELIANCE.NS"
6                    ,start=start_date,end=end_date
7                    ,quote="AdjClose",provider = "yahoo")
8 nifty<-get.hist.quote(instrument = "NSEI"
9                      ,start=start_date,end=end_date
10                     ,quote="AdjClose",provider = "yahoo")
11 data <-merge(nifty,rel)
12 rt<-diff(log(data))
13 head(rt*100)
```

The terminal output shows the following message:



```
R 4.2.2 - :~/Downloads/Teaching/Regression_and_Classification/NPTEL/
computational_finance.

See 'library(help="tseries")' for details.

> Sys.Date()
[1] "2023-02-07"
```

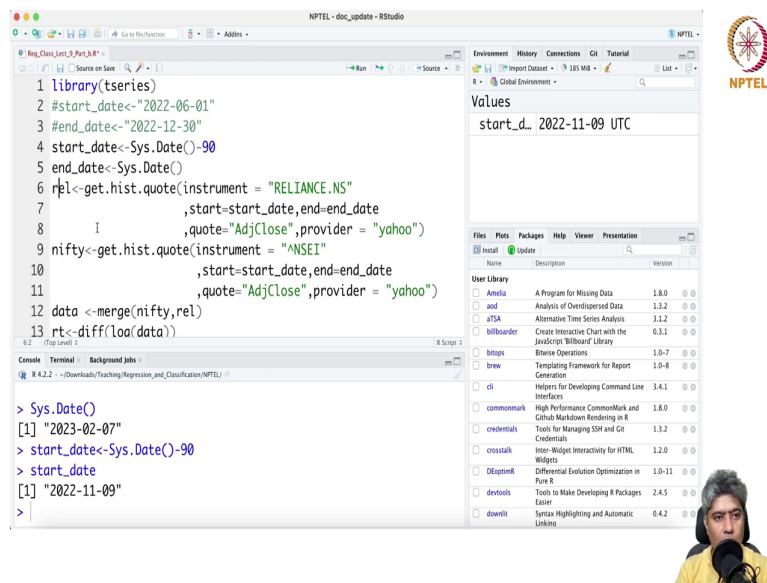
The Environment pane on the right shows "Environment is empty". The Packages pane shows the installed user library.

Name	Description	Version
Amelia	A Program for Missing Data	1.8.0
and	Analysis of Overdispersed Data	1.1.2
atsa	Alternative Time Series Analysis	3.1.2
billboarder	Create Interactive Chart with the JavaScript 'Billboard' Library	0.3.1
bitops	Bitwise Operations	1.0-7
brew	Templating Framework for Report Generation	1.0-8
cli	Widgets for Developing Command Line Interfaces	3.4.1
commonmark	High Performance CommonMark and GitHub Markdown Rendering in R	1.8.0
credentials	Tools for Managing SSH and Git Credentials	1.3.2
crossstalk	Inter-Widget Interactivity for HTML Widgets	1.2.0
DEoptimR	Differential Evolution Optimization in Pure R	1.0-11
devtools	Tools to Make Developing R Packages Easier	2.4.5
downlit	Smarty Highlighting and Automatic Linking	0.4.2



However, what we can do? We can instead of that we can take the ended as usual some ended of Sys dot Date if you just take Sys dot Date. So, that will give you the today's date, today is 7th February 2023 as of I am recording this video it is 7th February 2023.

(Refer Slide Time: 02:38)



The screenshot displays the RStudio interface with the following R code in the editor:

```
1 library(tseries)
2 #start_date<-"2022-06-01"
3 #end_date<-"2022-12-30"
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rpl<-get.hist.quote(instrument = "RELIANCE.NS"
7                     ,start=start_date,end=end_date
8                     ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "NSEI"
10                      ,start=start_date,end=end_date
11                      ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
```

The console output shows the following commands and results:

```
> Sys.Date()
[1] "2023-02-07"
> start_date<-Sys.Date()-90
> start_date
[1] "2022-11-09"
>
```

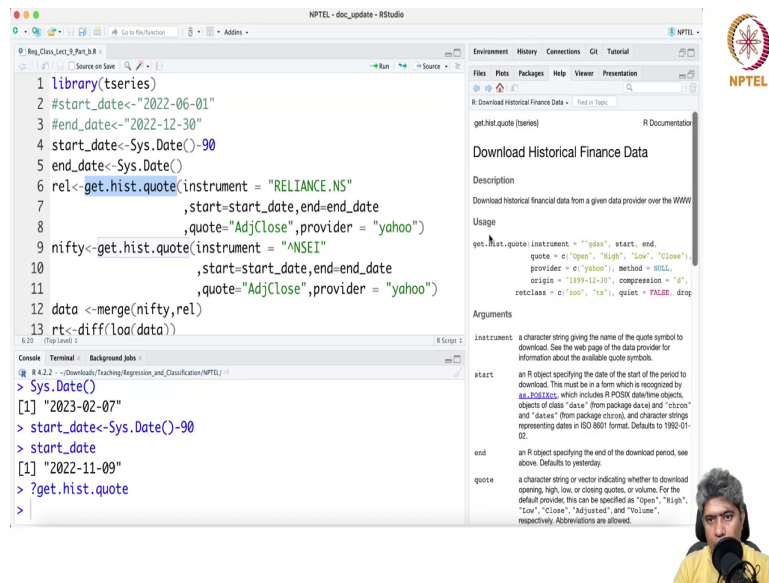
The right-hand pane shows the Environment tab with the following values:

Variable	Value
start_d_	2022-11-09 UTC

The Packages pane lists installed and available packages, including the 'tseries' package.

And what I can do is I can take a start date which is maybe 60 days prior to the system date. So, what I can do? I can just take Sys Date minus 180 days like 6 or maybe I just I will just take last 90 days of data. So, if I just give this then start date will be 90 days prior which is 9th November 2022. Now, given these 2 date. So, now, get dot hist dot quote is the function that comes with the package tseries.

(Refer Slide Time: 03:23)





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

```
1 library(tseries)
2 #start_date<-"2022-06-01"
3 #end_date<-"2022-12-30"
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
7                   ,start=start_date,end=end_date
8                   ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "NSEI"
10                    ,start=start_date,end=end_date
11                    ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
```

The console window shows the execution of the code:

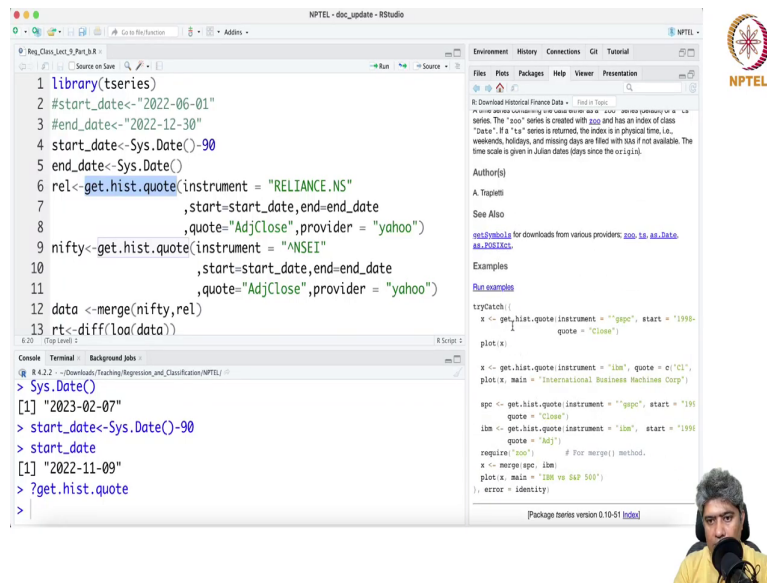
```
> Sys.Date()
[1] "2023-02-07"
> start_date<-Sys.Date()-90
> start_date
[1] "2022-11-09"
> ?get.hist.quote
>
```

The right-hand pane shows the help documentation for the `get.hist.quote` function. The title is "Download Historical Finance Data". The description states: "Download historical financial data from a given data provider over the WWW". The usage is: `get.hist.quote(instrument = "idst", start, end, quote = c("Open", "High", "Low", "Close"), provider = c("yahoo"), method = "URL", origin = "1899-12-30", compression = "G", retclass = c("foo", "ts"), quiet = FALSE, drop)`. The arguments section explains the parameters: `instrument` is a character string giving the name of the quote symbol; `start` is an R object specifying the date of the start of the period to download; `end` is an R object specifying the end of the download period; and `quote` is a character string or vector indicating whether to download opening, high, low, or closing quotes, or volume.



So, if you just question mark get dot hist dot quote then you can download historical financial data ok and here there are some examples also.

(Refer Slide Time: 03:34)



The screenshot displays the RStudio interface with the following R code in the editor:

```
1 library(tseries)
2 #start_date<-"2022-06-01"
3 #end_date<-"2022-12-30"
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
7                    ,start=start_date,end=end_date
8                    ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "NSEI"
10                     ,start=start_date,end=end_date
11                     ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
```

The console shows the execution of the first few lines of code:

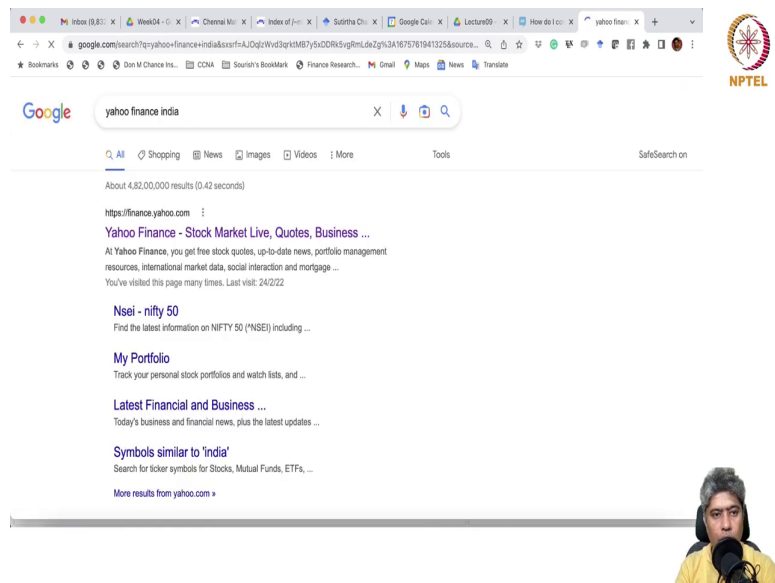
```
R 4.2.2 - [D:\Downloads\Teaching\Regression_and_Classification\NPTEL]
> Sys.Date()
[1] "2023-02-07"
> start_date<-Sys.Date()-90
> start_date
[1] "2022-11-09"
> ?get.hist.quote
```

The right-hand pane shows the documentation for the `get.hist.quote` function, including a description of the `zoo` package and examples of how to use the function for different instruments like 'ibm' and 'spc'.

A small video inset in the bottom right corner shows a person with a microphone, likely the presenter, speaking.

So, what you can do? I am going to download the instrument name RELIANCE dot NS. You can try some other stock also like TCS or something, but you have to know with the name of the instrument in this case TCS.

(Refer Slide Time: 04:03)



So, one way to figure out which instrument you want to use is you go to Google Yahoo you search Yahoo finance India make sure you are giving Yahoo finance India and then you if you click on that.

(Refer Slide Time: 04:20)

The screenshot shows the Yahoo Finance website interface. At the top, there is a navigation bar with links for Home, Mail, News, Finance, Sports, Entertainment, Search, Mobile, and More. Below this is a search bar and a 'Sign In' button. The main content area features a market overview section with various indices and commodities, including S&P Futures, Dow Futures, Nasdaq Futures, Russell 2000 Futures, Crude Oil, and Gold. A large news article is prominently displayed with the headline 'Stocks moving after hours: Chegg, Blizzard, more'. To the right, there is a 'My Portfolio & Markets' section with a search bar and a table of recently viewed assets. A small video thumbnail of a person wearing a hat and headphones is visible in the bottom right corner of the screenshot.

Index	Value	Change
S&P Futures	4,131.25	+17.50 (+0.43%)
Dow Futures	33,948.00	+144.00 (+0.42%)
Nasdaq Futures	12,561.25	+45.50 (+0.37%)
Russell 2000 Futures	1,968.40	+9.30 (+0.47%)
Crude Oil	75.67	+0.50 (+0.67%)
Gold	1,888.70	+9.20 (+0.49%)

U.S. markets open in 5 hours 4 minutes

Stocks moving after hours: Chegg, Blizzard, more

Shares of these companies are making the biggest moves in after-hours trading

Boeing to slash 2,000 white-collar jobs in finance, HR

Reuters

Yellen: 'You don't have a recession' with unemployment so low

Reuters

Stocks close lower as investors ponder Fed's rate path

Yahoo Finance

Symbol	Last Price	Change	% Change
NYSE	12,777.00	+12.40	+0.07%
NFTY 50			

And then you can look for typically quote look up here if you just gives a TCS.

(Refer Slide Time: 04:33)

The screenshot shows the Yahoo Finance website with the NIFTY 50 index highlighted. The index value is 17,773.75, up 9.15 (+0.05%) from the previous close. A video inset in the bottom right corner shows a man with a beard and a yellow jacket speaking into a microphone. The NPTEL logo is visible in the top right corner.

Instrument	Price	Change
S&P Futures	4,130.50	+7.06 (+0.17%)
Dow Futures	33,946.00	+23.90 (+0.07%)
Nasdaq Futures	12,558.75	+43.29 (+0.35%)
Russell 2000 Futures	1,968.20	+2.10 (+0.11%)
Crude Oil	75.62	+1.11 (+1.48%)
Gold	1,888.50	+8.80 (+0.46%)

Summary	Chart	Conversations	Historical Data	Options	Components							
Previous Close	17,764.60	Day's Range	17,652.55 - 17,811.15	1D	5D	1M	6M	YTD	1Y	5Y	Max	Full screen
Open	17,790.10	52 Week Range	15,183.40 - 18,887.60									
Volume	0	Avg. Volume	252,771									

If you gives TCS ok.

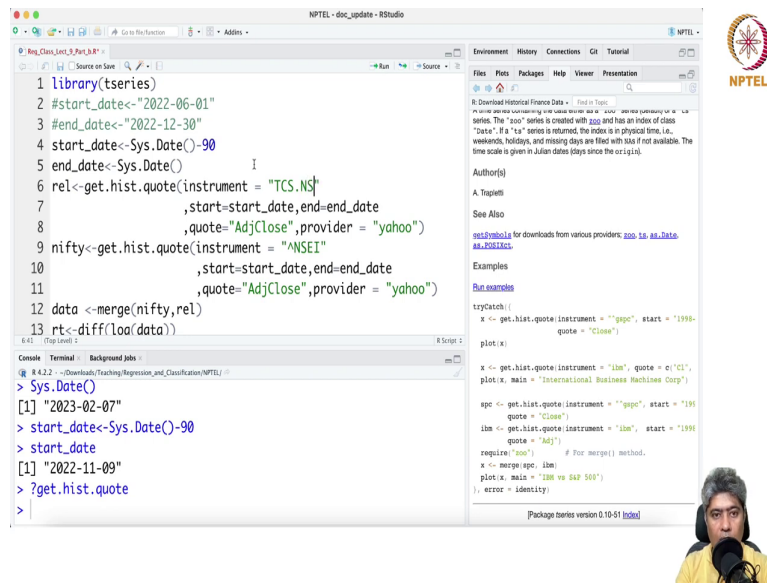
(Refer Slide Time: 04:36)

The screenshot shows the Yahoo Finance website interface. At the top, there are navigation links for Finance Home, Watchlists, My Portfolio, Crypto, Yahoo Finance Plus, News, Screeners, Markets, Videos, and Personal Finance. Below this, there are market indicators for S&P Futures, Dow Futures, Nasdaq Futures, Russell 2000 Futures, Crude Oil, and Gold. The main focus is on Tata Consultancy Services Limited (TCS.NS), with a current price of 3,486.20 and a daily change of +26.25 (+0.76%). A table of key statistics is provided, including Previous Close, Market Cap, Beta, PE Ratio, and EPS. A small video inset of a person is visible in the bottom right corner of the screenshot.

Key Metric	Value
Previous Close	3,459.95
Market Cap	12,756T
Open	3,468.90
Bid	3,486.10 x 0
Ask	3,486.30 x 0
Day's Range	3,461.40 - 3,493.00
Market Cap	12,756T
Beta (5Y Monthly)	0.58
PE Ratio (TTM)	31.44
EPS (TTM)	110.90
Earnings Date	Apr 10, 2023 - Apr 14, 2023

So, TCS dot NS is the Tata Consultancy Services. So, Tata Consultancy Services that is TCS dot NS. So, if I want to download the data for Tata Consultancy Services. So, I will choose TCS dot NS and I will go in my R and here instead of RELIANCE dot is.

(Refer Slide Time: 04:57)



```
1 library(tseries)
2 #start_date<-"2022-06-01"
3 #end_date<-"2022-12-30"
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "TCS.NS",
7                    ,start=start_date,end=end_date
8                    ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "ANSEI"
10                    ,start=start_date,end=end_date
11                    ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
```


```
> Sys.Date()
[1] "2023-02-07"
> start_date<-Sys.Date()-90
> start_date
[1] "2022-11-09"
> ?get.hist.quote
```

NPTEL

Download Historical Finance Data

trycatch({
 x <- get.hist.quote(instrument = "SPC", start = "1998-
 quote = "Close")
 plot(x)
 x <- get.hist.quote(instrument = "IBM", quote = "CL",
 plot(x, main = "International Business Machine Corp")
 spc <- get.hist.quote(instrument = "SPC", start = "1998
 quote = "Close")
 ibm <- get.hist.quote(instrument = "IBM", start = "1998
 quote = "Adj")
 require("zoo") # For merge() method.
 x <- merge(spc, ibm)
 plot(x, main = "IBM vs S&P 500")
 }, error = identity)

Package zoo version 0.10-51 [Index](#)



I will say TCS dot NS, but I will just keep it is as Reliance dot NS.

(Refer Slide Time: 05:07)

```
1 library(tseries)
2 #start_date<-"2022-06-01"
3 #end_date<-"2022-12-30"
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
7                    ,start=start_date,end=end_date
8                    ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "ANSEI"
10                     ,start=start_date,end=end_date
11                     ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
```

Console:

```
R 4.2.2 -> Downloads/Teaching/Regression_and_Classification/NPTEL/
[1] "2022-11-09"
> ?get.hist.quote
> rel<-get.hist.quote(instrument = "RELIANCE.NS"
+                    ,start=start_date,end=end_date
+                    ,quote="AdjClose",provider = "yahoo")
Error in as.Date(end) : object 'end_date' not found
>
```

Environment: R 4.2.2 (2022-10-31)

Files: Files Packages Help Viewer Presentation

Download Historical Finance Data - find in Topic

Author(s): A. Taglietti

See Also: getSymbols for downloads from various providers: [getSymbols](#), [getSymbols](#), [getSymbols](#)

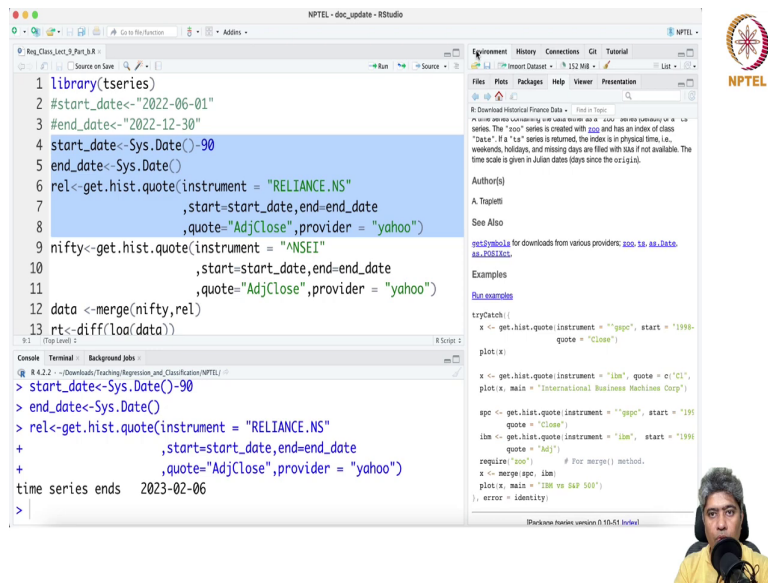
Examples: Run examples

```
tryCatch({
  x <- get.hist.quote(instrument = "SPX", start = "1998-01-01",
                    quote = "Close")
  plot(x)
}, error = identity)
```

Package: tseries version 0.10-51 [Index](#)

And I will just download the data oh I need to just run this to and I will just run the data.

(Refer Slide Time: 05:12)



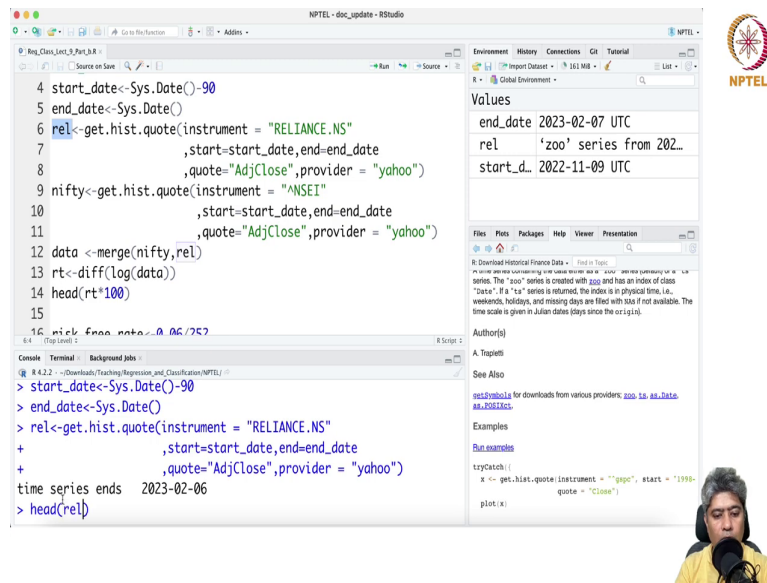
```
1 library(tseries)
2 #start_date<-"2022-06-01"
3 #end_date<-"2022-12-30"
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
7                   ,start=start_date,end=end_date
8                   ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "NSEI"
10                    ,start=start_date,end=end_date
11                    ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
```

```
> start_date<-Sys.Date()-90
> end_date<-Sys.Date()
> rel<-get.hist.quote(instrument = "RELIANCE.NS"
+                   ,start=start_date,end=end_date
+                   ,quote="AdjClose",provider = "yahoo")
time series ends 2023-02-06
```

The screenshot also shows the RStudio interface with a terminal window at the bottom and a help pane on the right. The help pane displays information for the 'get.hist.quote' function, including its author (A. Tognetti) and examples of usage.



(Refer Slide Time: 05:15)



The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for downloading historical data from Yahoo Finance and merging it with Nifty data. The code includes:

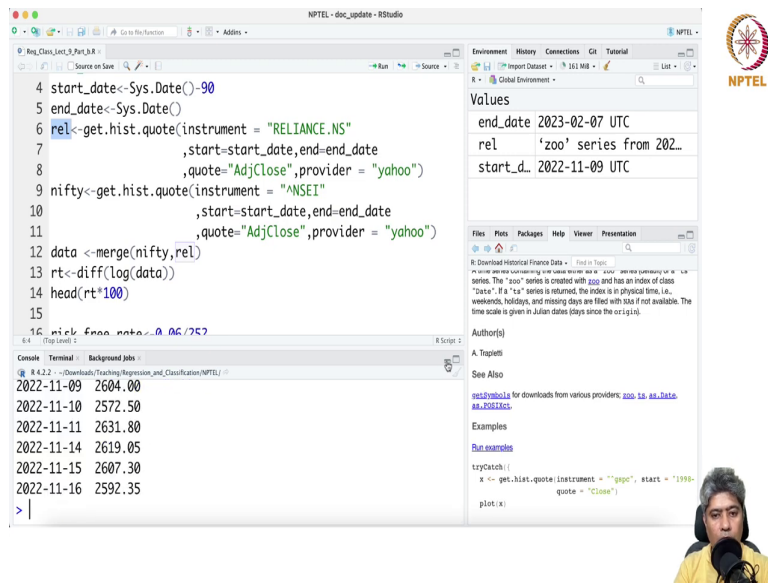
```
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
7                     ,start=start_date,end=end_date
8                     ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "NSEI"
10                      ,start=start_date,end=end_date
11                      ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
14 head(rt*100)
15
16 nick finn notes 0 AG / 252
```
- Console:** Shows the execution of the code, including the output of the `head` function:

```
> start_date<-Sys.Date()-90
> end_date<-Sys.Date()
> rel<-get.hist.quote(instrument = "RELIANCE.NS"
+                     ,start=start_date,end=end_date
+                     ,quote="AdjClose",provider = "yahoo")
time series ends 2023-02-06
> head(rel)
```
- Environment Pane:** Shows the 'Values' section with the following information:

```
end_date 2023-02-07 UTC
rel 'zoo' series from 202...
start_d_ 2022-11-09 UTC
```
- Help Pane:** Displays the documentation for the `get.hist.quote` function, including the author's name (A. Tajtehi) and a link to the `getSymbols` function for downloading data from various providers.

Now, if you go to the environment. So, you can see the Reliance it has downloaded as zoo package. Similarly, you have if you just. So, if you just say head Reliance this is the object name if you just say Reliance.

(Refer Slide Time: 05:37)



The screenshot displays the RStudio interface with the following R code in the script editor:

```
4 start_date<-Sys.Date()-90
5 end_date<-Sys.Date()
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
7                   ,start=start_date,end=end_date
8                   ,quote="AdjClose",provider = "yahoo")
9 nifty<-get.hist.quote(instrument = "ANSEI"
10                    ,start=start_date,end=end_date
11                    ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
14 head(rt*100)
15
16 nick finn notes 0 06 / 252
```

The console output shows the first six rows of the calculated returns:

```
2022-11-09 2604.00
2022-11-10 2572.50
2022-11-11 2631.80
2022-11-14 2619.05
2022-11-15 2607.30
2022-11-16 2592.35
```

The right-hand pane shows the 'Values' section with the following data:

Variable	Value
end_date	2023-02-07 UTC
rel	'zoo' series from 202...
start_d	2022-11-09 UTC

The right-hand pane also includes a 'Download Historical Finance Data' section with the following text:

Download Historical Finance Data - Find in Topic
This tool serves as a wrapper for the `get.hist.quote` function. It allows users to specify the instrument, start and end dates, and the provider. The 'zoo' series is created with `zoo` and has an index of class 'Date'. If a 'tz' series is returned, the index is in physical time, i.e., weekends, holidays, and missing days are filled with `NA` if not available. The time scale is given in Julian dates (days since the origin).

Author(s)
A. Taglietti
See Also
[getSymbols](#) for downloads from various providers; [zoo](#); [tz](#); [as.Date](#); [as.POSIXct](#).
Examples
[Run example](#)

```
trycatch({
  x <- get.hist.quote(instrument = "AAPL", start = "1999-01-01",
                    quote = "Close")
  plot(x)
```

The NPTEL logo is visible in the top right corner, and a small video thumbnail of a person is in the bottom right corner.

(Refer Slide Time: 05:39)

```
NPTEL - doc_update - RStudio
Source
Environment History Connections Git Tutorial
R 4.2.2 -- Downloads/Teaching/Regression_and_Classification/NPTEL/
[1] "2022-11-09"
> ?get.hist.quote
> rel<-get.hist.quote(instrument = "RELIANCE.NS"
+                      ,start=start_date,end=end_date
+                      ,quote="AdjClose",provider = "yahoo")
Error in as.Date(end) : object 'end_date' not found
> start_date<-Sys.Date()-90
> end_date<-Sys.Date()
> rel<-get.hist.quote(instrument = "RELIANCE.NS"
+                      ,start=start_date,end=end_date
+                      ,quote="AdjClose",provider = "yahoo")
time series ends 2023-02-06
> head(rel)
              Adjusted
2022-11-09  2604.00
2022-11-10  2572.50
2022-11-11  2631.80
2022-11-14  2619.05
2022-11-15  2607.301
2022-11-16  2592.35
> tail(rel)
```

Values

end_date	2023-02-07 UTC
rel	'zoo' series from 202...
start_d_	2022-11-09 UTC

Files Plots Packages Help Viewer Presentation

R: Download Historical Finance Data - Plot in Topic

Printed series containing the close price of the 'zoo' series containing the 'zoo' series. The 'zoo' series is created with zoo and has an index of class 'Date'. If a 'zoo' series is returned, the index is in physical time, i.e., weekends, holidays, and missing days are filled with NA if not available. The time scale is given in Julian dates (days since the origin).

Author(s)

A. Tapani

See Also

seeSymbols for downloads from various providers: zoo, ts, as.Date, as.POSIXct.

Examples

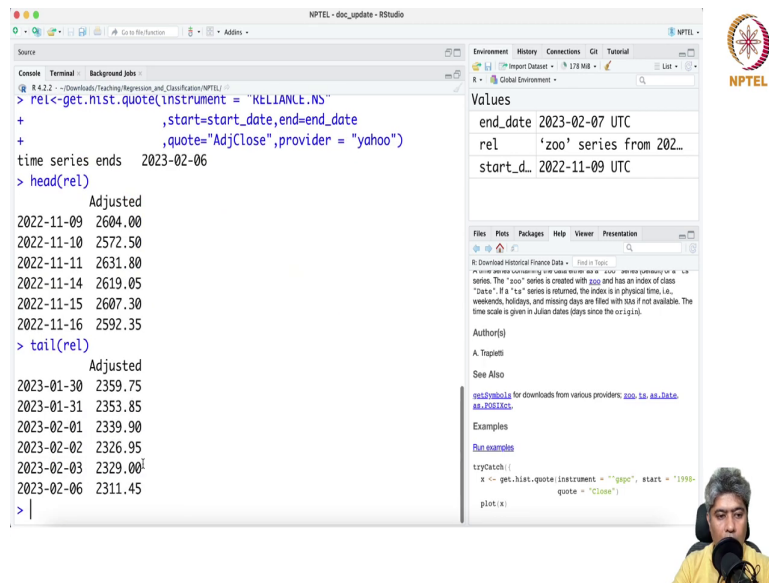
Run example

```
trycatch({
  x = get.hist.quote(instrument = "AAPL", start = "1999-01-01",
                    quote = "Close")
  plot(x)
```

So, it has the adjusted close prices of Reliance.

Now, if you say tail of TL.

(Refer Slide Time: 05:51)



The screenshot displays the RStudio interface. The console shows the following R code and output:

```
> rel<-get.hist.quote(instrument = "NIFTY.NS",
+                    ,start=start_date,end=end_date
+                    ,quote="AdjClose",provider = "yahoo")
time series ends 2023-02-06
> head(rel)
      Adjusted
2022-11-09 2604.00
2022-11-10 2572.50
2022-11-11 2631.80
2022-11-14 2619.05
2022-11-15 2607.30
2022-11-16 2592.35
> tail(rel)
      Adjusted
2023-01-30 2359.75
2023-01-31 2353.85
2023-02-01 2339.90
2023-02-02 2326.95
2023-02-03 2329.00
2023-02-06 2311.45
> |
```

The right-hand pane shows the 'Values' table:

end_date	rel	start_d_
2023-02-07 UTC	'zoo' series from 202...	2022-11-09 UTC

Below the table, there is a help window for the `get.hist.quote` function, showing the author (A. Taplett) and a trycatch block:

```
trycatch(
  x = get.hist.quote(instrument = "AAPL", start = "1999-
  quote = "Close")
  plot(x)
```

So, till yesterday the price was 2311.45 per share ok. Now, nifty is the index if you go back if you remember.

(Refer Slide Time: 06:04)

The image shows a screenshot of the Yahoo Finance website displaying the Nifty 50 index. The browser's address bar shows the URL 'finance.yahoo.com/quote/%5ENIFTY50:FIN%5ENSEI'. The page header includes the Yahoo Finance logo and a search bar. Below the header, there are several market indicators: S&P 500 (4,130.50, +7.86 (+0.17%)), Dow Jones (33,946.00, +12.80 (+0.04%)), Nikkei 225 (12,559.25, +43.71 (+0.33%)), Russell 2000 (1,968.00, +2.90 (+0.15%)), Crude Oil (75.53, +1.42 (+1.92%)), and Gold (1,889.10, +8.60 (+0.45%)).

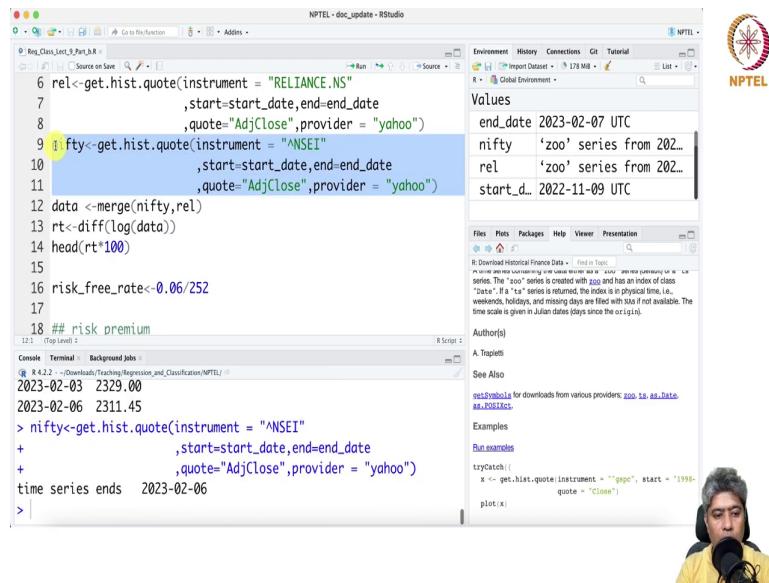
The main section is titled 'NIFTY 50 (^NSEI)' and shows a current price of 17,771.65, up 7.05 (+0.04%). Below this, there is a 'Summary' tab and a table of key statistics:

Previous Close	Day's Range	1D	5D	1M	6M	YTD	1Y	5Y	Max
17,764.60	17,692.55 - 17,811.55								
Open	52 Week Range	1D	5D	1M	6M	YTD	1Y	5Y	Max
17,790.10	15,183.40 - 18,887.60								
Volume	Avg. Volume	1D	5D	1M	6M	YTD	1Y	5Y	Max
0	252,771								

Below the table is a candlestick chart showing price movement from 10 AM to 02 PM. A tooltip over the chart reads: 'NEW: Experience our best charts yet. Everything you need for any kind of investing.' At the bottom of the screenshot, there is a news headline: 'India WPI inflation eases more than expected in November' by Anshul Wajrick, dated 11/03/2023. A video call overlay of a person in a yellow shirt is visible in the bottom right corner.

So, this is the NIFTY 50, NIFTY 50 is the index of National Stock Exchange this is the 50 largest company of the country.

(Refer Slide Time: 06:21)



The screenshot displays the RStudio interface with the following R code in the editor:

```
6 rel<-get.hist.quote(instrument = "RELIANCE.NS"  
7 ,start=start_date,end=end_date  
8 ,quote="AdjClose",provider = "yahoo")  
9 nifty<-get.hist.quote(instrument = "NSEI"  
10 ,start=start_date,end=end_date  
11 ,quote="AdjClose",provider = "yahoo")  
12 data <-merge(nifty,rel)  
13 rt<-diff(log(data))  
14 head(rt*100)  
15  
16 risk_free_rate<-0.06/252  
17  
18 ## risk premium
```

The console output shows the execution of the NIFTY 50 data fetch command:

```
R 4.2.2 ~> Downloads/Teaching/Regression_and_Classification/NPTEL/ <-->  
2023-02-03 2329.00  
2023-02-06 2311.45  
> nifty<-get.hist.quote(instrument = "NSEI"  
+ ,start=start_date,end=end_date  
+ ,quote="AdjClose",provider = "yahoo")  
time series ends 2023-02-06  
>
```

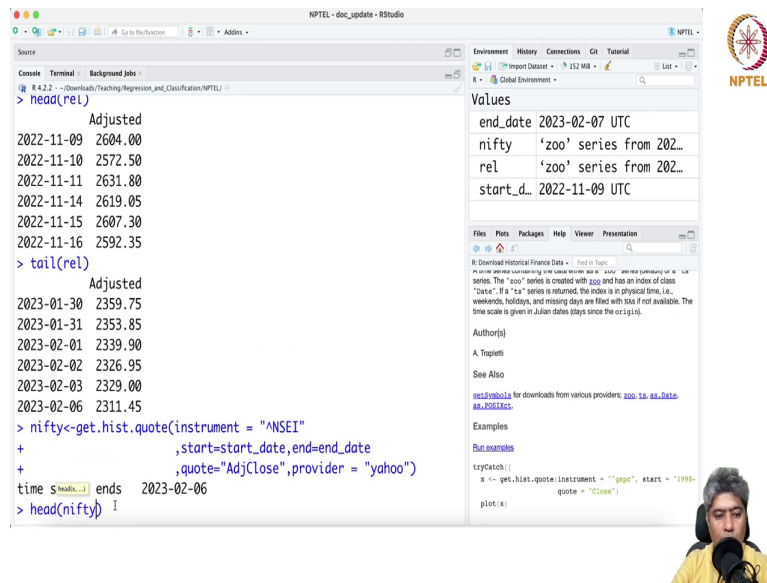
The right-hand pane shows the 'Values' section with the following data:

end_date	start_d...
2023-02-07 UTC	2022-11-09 UTC

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

So, if you just run the NIFTY 50 it will give you the NIFTY values also.

(Refer Slide Time: 06:29)



The screenshot shows the RStudio interface with the following content:

```
Source
Console Terminal Background jobs
R 4.2.2 - Downloads/Teaching/Regression_and_Classification/NPTEL/
> head(rel)
Adjusted
2022-11-09 2604.00
2022-11-10 2572.50
2022-11-11 2631.80
2022-11-14 2619.05
2022-11-15 2607.30
2022-11-16 2592.35
> tail(rel)
Adjusted
2023-01-30 2359.75
2023-01-31 2353.85
2023-02-01 2339.90
2023-02-02 2326.95
2023-02-03 2329.00
2023-02-06 2311.45
> nifty<-get.hist.quote(instrument = "NSEI"
+                       ,start=start_date,end=end_date
+                       ,quote="AdjClose",provider = "yahoo")
time s head(x) ends 2023-02-06
> head(nifty) I
```

Values

end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
start_d_	2022-11-09 UTC

Files Plots Packages Help Viewer Presentation

R: Download Historical Finance Data - Plot in Topic

Printed series containing the close price of the NSEI series. The "zoo" series is created with zoo and has an index of class "Date". If a "ts" series is returned, the index is in physical time, i.e., weekends, holidays, and missing days are filled with NA if not available. The time scale is given in Julian dates (days since the origin).

Author(s)



A. Tognetti

See Also

[getSymbols](#) for downloads from various providers: [zoo](#), [ts](#), [aa](#), [haha](#), [aa](#), [R2013fct](#).

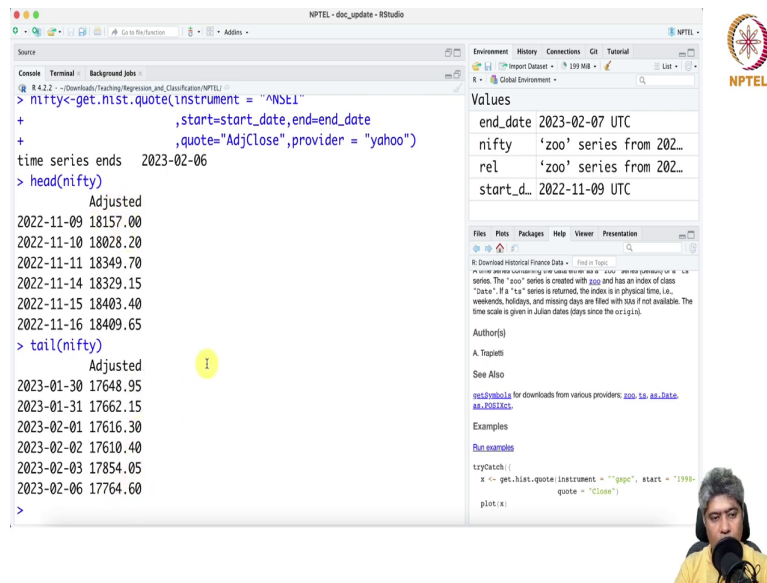
Examples

```
trycatch({
  x <- get.hist.quote(instrument = "AAPL", start = "1999-
  quote = "Close")
plot(x)
```



Similarly, you can see the you can say head of nifty.

(Refer Slide Time: 06:35)



```
> nifty<-get.hist.quote(instrument = "NSEI"  
+                       ,start=start_date,end=end_date  
+                       ,quote="AdjClose",provider = "yahoo")  
time series ends 2023-02-06  
> head(nifty)  
      Adjusted  
2022-11-09 18157.00  
2022-11-10 18028.20  
2022-11-11 18349.70  
2022-11-14 18329.15  
2022-11-15 18403.40  
2022-11-16 18409.65  
> tail(nifty)  
      Adjusted  
2023-01-30 17648.95  
2023-01-31 17662.15  
2023-02-01 17616.30  
2023-02-02 17610.40  
2023-02-03 17854.05  
2023-02-06 17764.60  
>
```

end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
start_d_	2022-11-09 UTC

Description
Download Historical Finance Data
A time series containing the close price of the NIFTY 50 index. The "zoo" series is created with zoo and has an index of class "Date". If a "ts" series is returned, the index is in physical time, i.e., weekends, holidays, and missing days are filled with NA if not available. The time scale is given in Julian dates (days since the origin).

Author(s)
A. Tognetti
See Also
[getSymbols](#) for downloads from various providers; [zoo.ts.ts.date.ts.objects](#)
Examples
[Run example](#)
trycatch(
 x = get.hist.quote(instrument = "NSEI", start = "1995-01-01", end = "2023-02-06", quote = "Close")
 plot(x)

And similarly, you can just say tail of nifty. You can see that in November it was running around 18,000 now it is near came down to 17,000 the market is not good now as of February.

(Refer Slide Time: 06:53)

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

```
9 nifty<-get.hist.quote(instrument = "NSEI"  
10                       ,start=start_date,end=end_date  
11                       ,quote="AdjClose",provider = "yahoo")  
12 data <-merge(nifty,rel)  
13 rt<-diff(log(data))  
14 head(rt*100)  
15  
16 risk_free_rate<-0.06/252  
17  
18 ## risk premium  
19 rt<-rt-risk_free_rate  
20
```

The console shows the execution of the merge function:

```
> data <- merge(nifty, rel)  
> head(data)
```

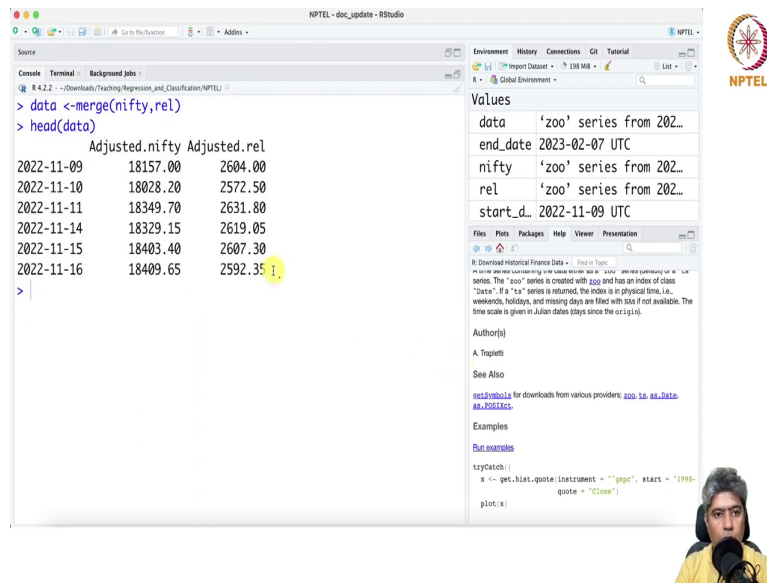
The Environment pane on the right shows the following variables:

Variable	Value
data	'zoo' series from 202...
end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
start_d_	2022-11-09 UTC

A small video inset in the bottom right corner shows a person speaking into a microphone.

So, now what I am going to do? I am going to run the merge these two and it will automatically merge and so, head data.

(Refer Slide Time: 07:07)



The screenshot shows the RStudio interface. The console window displays the following R code and output:

```
> data <- merge(nifty, rel)
> head(data)
      Adjusted.nifty Adjusted.rel
2022-11-09    18157.00    2604.00
2022-11-10    18028.20    2572.50
2022-11-11    18349.70    2631.80
2022-11-14    18329.15    2619.05
2022-11-15    18403.40    2607.30
2022-11-16    18409.65    2592.35
```

The help window for the 'get.hist.quote' function is open, showing the following information:

Values

data	'zoo' series from 202...
end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
start_d_	2022-11-09 UTC

Author(s)
A. Taglietti

See Also
[getSymbols](#) for downloads from various providers; [get.hist.quote](#); [get.hist.quote](#); [get.hist.quote](#)

Examples
[Run example](#)

```
tryCatch({
  x <- get.hist.quote(instrument = "AAPL", start = "1999-01-01",
    quote = "close")
  plot(x)
```

So, now it is a Adjusted nifty and Adjusted Reliance. So, two series got merged by date automatically. So, you do not have to worry about that.

(Refer Slide Time: 07:23)

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

```
10     ,start=start_date,end=end_date
11     ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
14 head(rt*100)
15
16 risk_free_rate<-0.06/252
17
18 ## risk premium
19 rt<-rt-risk_free_rate
20
21 plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'
```

The console output shows the following data:

Date	Adjusted.nifty	Adjusted.rel
2022-11-10	18028.20	2572.50
2022-11-11	18349.70	2631.80
2022-11-14	18329.15	2619.05
2022-11-15	18403.40	2607.70
2022-11-16	18409.65	2592.55

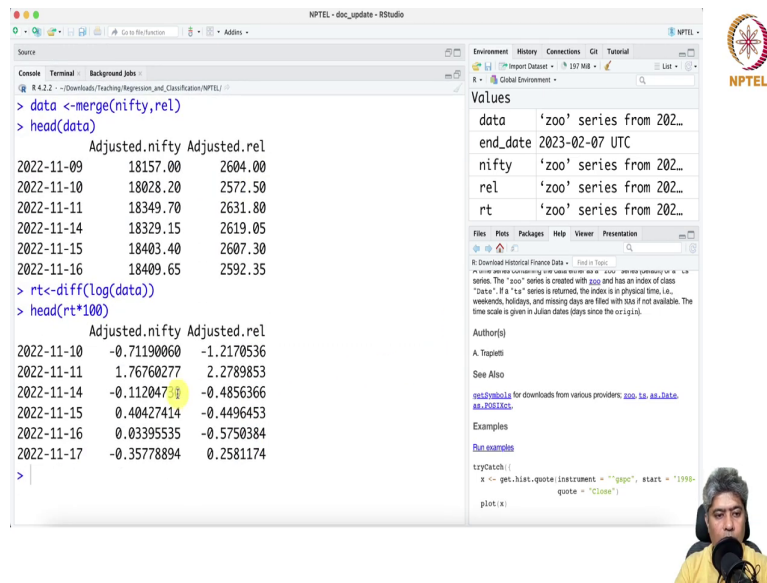
The console also shows the following commands and their outputs:

```
> rt<-head(log(data))
> head(rt*100)
```

The RStudio interface also shows the NPTEL logo and a small video feed of a person in the bottom right corner.

And now if you just calculate the this now head of rt times 100.

(Refer Slide Time: 07:36)



The screenshot shows the RStudio interface with the following code and output:

```
> data <- merge(nifty, rel)
> head(data)
      Adjusted.nifty Adjusted.rel
2022-11-09  18157.00    2604.00
2022-11-10  18028.20    2572.50
2022-11-11  18349.70    2631.80
2022-11-14  18329.15    2619.05
2022-11-15  18403.40    2607.30
2022-11-16  18409.65    2592.35
> rt <- diff(log(data))
> head(rt*100)
      Adjusted.nifty Adjusted.rel
2022-11-10  -0.71190060  -1.2170536
2022-11-11   1.76760277   2.2789853
2022-11-14  -0.11204700  -0.4856366
2022-11-15   0.40427414  -0.4496453
2022-11-16   0.03395535  -0.5750384
2022-11-17  -0.35778894   0.2581174
```

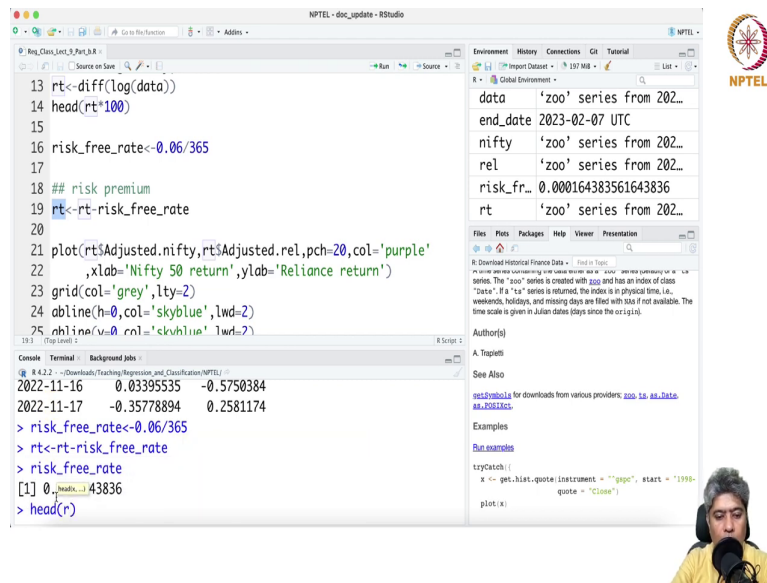
The right-hand pane shows the 'Values' section with the following information:

Variable	Value
data	'zoo' series from 202...
end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
rt	'zoo' series from 202...

A small video inset in the bottom right corner shows a person speaking into a microphone.

This is precisely the value that you have seen that this is the nifty log returns and these are the reliance log return. Ok. Now, I have taken risk free rate as 6 percent I am going to. So, that is why I have taken rate as 0.06 ok and I am going to divide that by 252 because there are 252 business days you can use 365 also, we can use 365 that will be probably more meaningful.

(Refer Slide Time: 08:10)



The screenshot shows an RStudio window with the following R code in the editor:

```
13 rt<-diff(log(data))
14 head(rt*100)
15
16 risk_free_rate<-0.06/365
17
18 ## risk premium
19 rt<-rt-risk_free_rate
20
21 plot(rtsAdjusted.nifty,rtsAdjusted.rel,pch=20,col='purple'
22      ,xlab='Nifty 50 return',ylab='Reliance return')
23 grid(col='grey',lty=2)
24 obline(h=0,col='skyblue',lwd=2)
25 obline(v=0,col='skvblue',lwd=2)
```

The console output shows the following results:

```
R 4.2.2 > ...Downloads/Teaching-Regression_and_Classification/NPTEL/...
2022-11-16 0.03395535 -0.5750384
2022-11-17 -0.35778894 0.2581174
> risk_free_rate<-0.06/365
> rt<-rt-risk_free_rate
> risk_free_rate
> risk_free_rate
[1] 0.0001643836
> head(r)
```

The Environment pane on the right shows the following objects:

Object	Value
data	'zoo' series from 202...
end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
risk_fr...	0.00016438361643836
rt	'zoo' series from 202...

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

So, we can use 365. So, the risk premium if we just calculate the subtract risk free from the return. So, now, it is risk premium.

(Refer Slide Time: 08:43)

The screenshot shows an RStudio session with the following code in the script editor:

```
13 rt<-diff(log(data))
14 head(rt*100)
15
16 risk_free_rate<-0.06/365
17
18 ## risk premium
19 rt<-rt-risk_free_rate
20
21 plot(rtsAdjusted.nifty, rtsAdjusted.rel, pch=20, col='purple'
22      , xlab='Nifty 50 return', ylab='Reliance return')
23 grid(col='grey', lty=2)
24 obline(h=0, col='skyblue', lwd=2)
25 obline(v=0, col='skyblue', lwd=2)
```

The console output shows the following data points:

```
2022-11-10 -0.0072833896 -0.012334919
2022-11-11  0.0175116442  0.022625469
2022-11-14 -0.0012848566 -0.005020750
2022-11-15  0.0038783578 -0.004660837
2022-11-16  0.0001751699 -0.005914767
2022-11-17 -0.0037422730  0.002416791
```

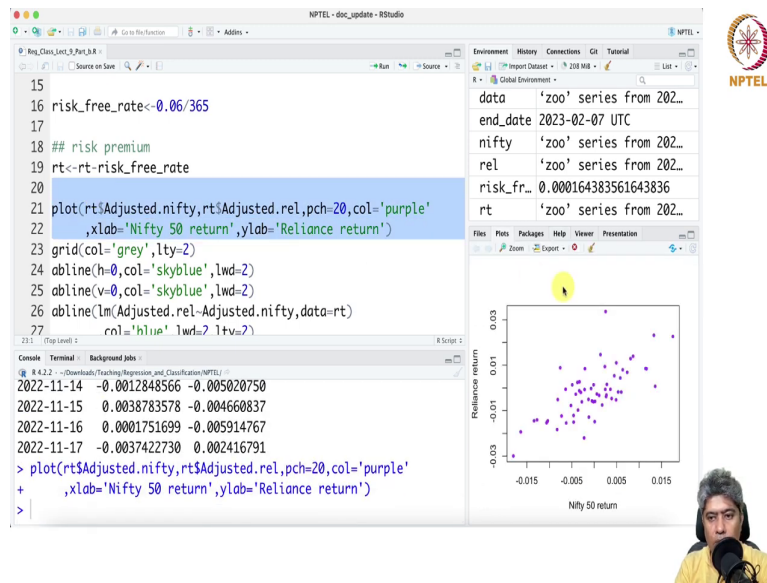
The Environment pane on the right shows the following objects:

Object	Class	Value
data	'zoo' series from 202...	
end_date	2023-02-07 UTC	
nifty	'zoo' series from 202...	
rel	'zoo' series from 202...	
risk_fr...	0.000164383561643836	
rt	'zoo' series from 202...	

The console also shows the output of the plot command: `plot.r`

Now RT because I have just subtracted the risk-free rate from all the returns and that; obviously, then the rate rt if data frame is now what I have is risk free rate risk premium So, these are all my risk premium now.

(Refer Slide Time: 08:48)



The screenshot displays the RStudio interface with the following R code in the script editor:

```
15  
16 risk_free_rate<-0.06/365  
17  
18 ## risk premium  
19 rt<-rt-risk_free_rate  
20  
21 plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
22 ,xlab='Nifty 50 return',ylab='Reliance return')  
23 grid(col='grey',lty=2)  
24 abline(h=0,col='skyblue',lwd=2)  
25 abline(v=0,col='skyblue',lwd=2)  
26 abline(lm(Adjusted.rel~Adjusted.nifty,data=rt)  
27 ,col='blue',lwd=2,lty=2)
```

The Environment pane on the right shows the following objects:

Object	Value
data	'zoo' series from 202...
end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
risk_fr...	0.000164383561643836
rt	'zoo' series from 202...

The Console shows the following output:

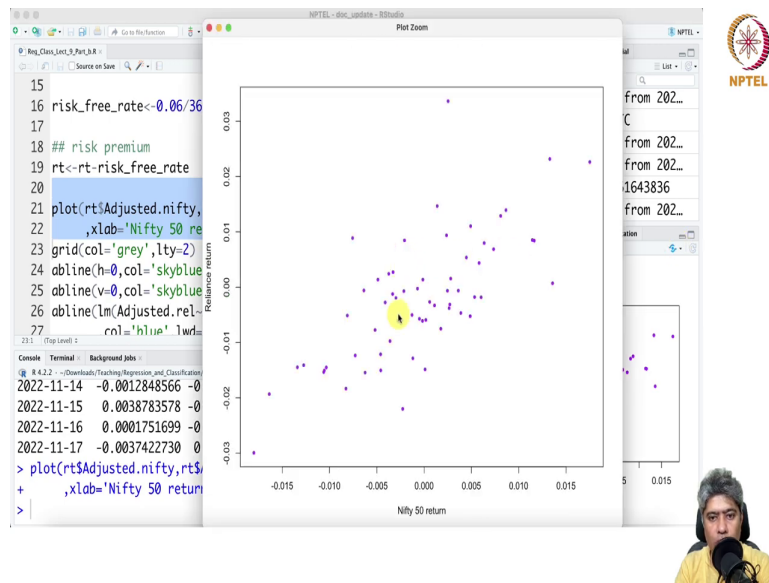
```
2022-11-14 -0.0012848566 -0.005020750  
2022-11-15 0.0038783578 -0.004660837  
2022-11-16 0.0001751699 -0.005914767  
2022-11-17 -0.0037422730 0.002416791  
> plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
+ ,xlab='Nifty 50 return',ylab='Reliance return')  
>
```

The plot shows a scatter plot of Adjusted Nifty 50 return (x-axis) versus Adjusted Reliance return (y-axis). The data points are purple circles. A blue regression line is fitted to the data. The x-axis ranges from -0.015 to 0.015, and the y-axis ranges from -0.03 to 0.03. A grid is visible in the background.

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

So, if I now plot risk premium that just it. So, these are my let me just zoom it.

(Refer Slide Time: 08:55)



So, these are the Nifty return on the x axis and Reliance return on the y axis ok looks like there is a or Nifty premium I should not say return Nifty premium risk premium and Reliance risk premium.

(Refer Slide Time: 09:14)

The screenshot displays the RStudio interface with the following R code in the editor:

```
15  
16 risk_free_rate<-0.06/365  
17  
18 ## risk premium  
19 rt<-rt-risk_free_rate  
20  
21 plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
22       ,xlab='Nifty 50 risk premium'  
23       ,ylab='Reliance risk premium')  
24 grid(col='grey',lty=2)  
25 abline(h=0,col='skyblue',lwd=2)  
26 abline(v=0,col='skyblue',lwd=2)  
27 ohline(lm(Adjusted.rel~Adjusted.nifty,data=rt))
```

The console shows the output of the `plot` function:

```
2022-11-17 -0.0037422730 0.002416791  
> plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
+       ,xlab='Nifty 50 return',ylab='Reliance return')  
> plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
+       ,xlab='Nifty 50 risk premium'  
+       ,ylab='Reliance risk premium')  
>
```

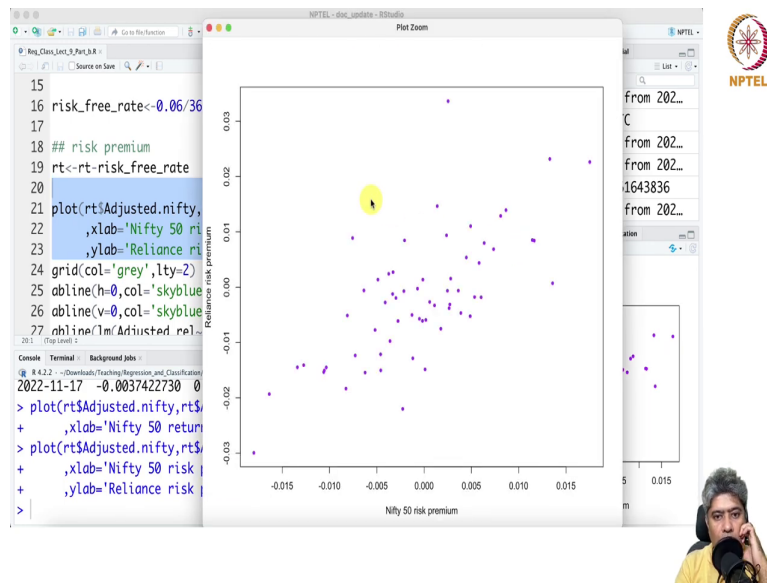
The environment pane on the right lists the following objects:

- data: 'zoo' series from 202...
- end_date: 2023-02-07 UTC
- nifty: 'zoo' series from 202...
- rel: 'zoo' series from 202...
- risk_fr...: 0.000164383561643836
- rt: 'zoo' series from 202...

The plot window shows a scatter plot with purple points (pch=20) representing the relationship between Nifty 50 risk premium (x-axis) and Reliance risk premium (y-axis). The plot includes a grey grid, a horizontal blue line at y=0, and a vertical blue line at x=0. A regression line is also visible. The NPTEL logo is present in the top right corner, and a small video inset of a person is in the bottom right corner.

So, that is the more correct appropriate value. So, let me just connect once more now it is corrected.

(Refer Slide Time: 09:33)



So, yeah Nifty risk premium and Reliance risk premium. So, now if I just run put some grid on it P9.

(Refer Slide Time: 09:48)

The screenshot displays the RStudio interface with the following R code in the editor:

```
15  
16 risk_free_rate<-0.06/365  
17  
18 ## risk premium  
19 rt<-rt-risk_free_rate  
20  
21 plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
22       ,xlab='Nifty 50 risk premium'  
23       ,ylab='Reliance risk premium')  
24 grid(col='grey',lty=2)  
25 abline(h=0,col='skyblue',lwd=2)  
26 abline(v=0,col='skyblue',lwd=2)  
27 abline(lm(Adjusted.rel~Adjusted.nifty,data=rt))
```



The console shows the execution of the plot command:

```
> plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'  
+       ,xlab='Nifty 50 risk premium'  
+       ,ylab='Reliance risk premium')  
> grid(col='grey',lty=2)  
> abline(h=0,col='skyblue',lwd=2)  
> abline(v=0,col='skyblue',lwd=2)  
>
```

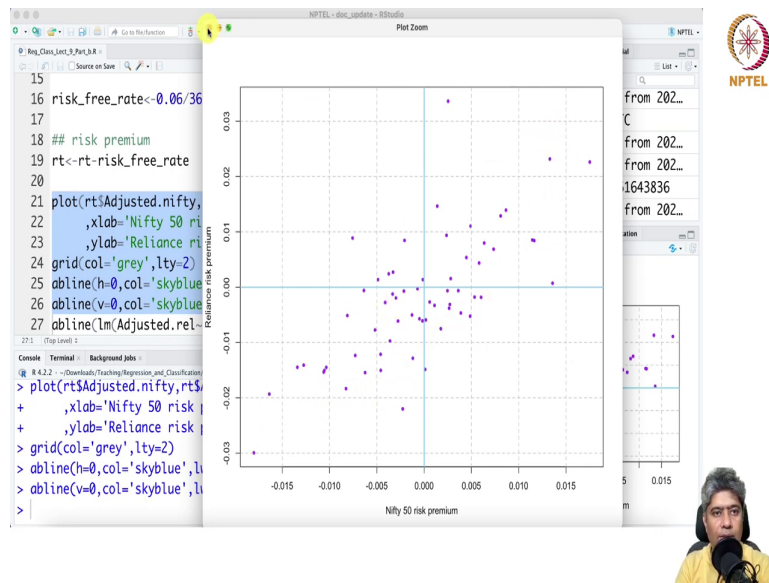
The environment pane on the right lists the following objects:

- data: 'zoo' series from 202...
- end_date: 2023-02-07 UTC
- nifty: 'zoo' series from 202...
- rel: 'zoo' series from 202...
- risk_fr...: 0.000164383561643836
- rt: 'zoo' series from 202...

The plot shows a scatter of purple points with a grey grid and two blue horizontal/vertical lines. The x-axis is labeled 'Nifty 50 risk premium' and the y-axis is labeled 'Reliance risk premium'. A regression line is also visible.

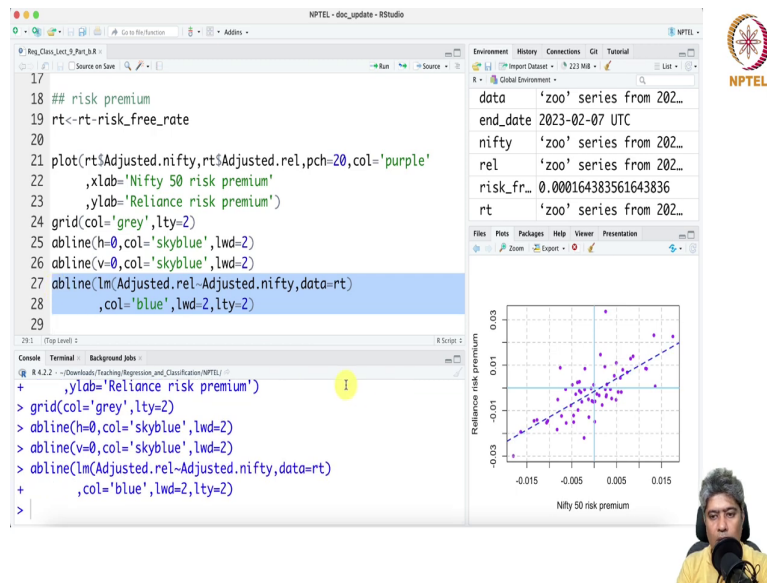


(Refer Slide Time: 09:51)



So, you can see around 0 I put a line and so, most of them are like in almost on a almost like a 45 degree line it is getting created.

(Refer Slide Time: 10:03)



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

```
17
18 ## risk premium
19 rt<-rt-risk_free_rate
20
21 plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'
22       ,xlab='Nifty 50 risk premium'
23       ,ylab='Reliance risk premium')
24 grid(col='grey',lty=2)
25 abline(h=0,col='skyblue',lwd=2)
26 abline(v=0,col='skyblue',lwd=2)
27 abline(lm(Adjusted.rel~Adjusted.nifty,data=rt)
28       ,col='blue',lwd=2,lty=2)
29
```

The console window shows the execution of the code:

```
29:1 (Top Level) >
R 4.2.2 > plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,col='purple'
+ ,ylab='Reliance risk premium')
> grid(col='grey',lty=2)
> abline(h=0,col='skyblue',lwd=2)
> abline(v=0,col='skyblue',lwd=2)
> abline(lm(Adjusted.rel~Adjusted.nifty,data=rt)
+ ,col='blue',lwd=2,lty=2)
>
```

The Environment pane on the right shows the following objects:

Object	Value
data	'zoo' series from 202...
end_date	2023-02-07 UTC
nifty	'zoo' series from 202...
rel	'zoo' series from 202...
risk_fr...	0.000164383561643836
rt	'zoo' series from 202...

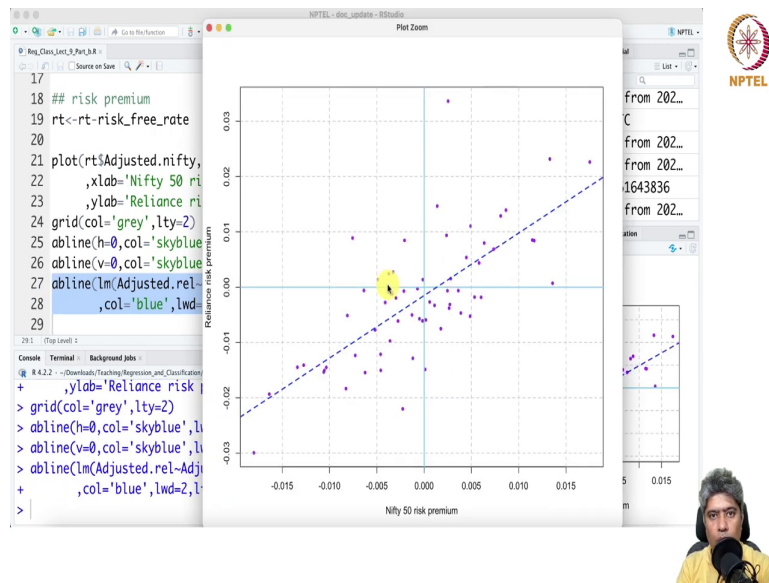
The Plots pane shows a scatter plot with the following axes:

- X-axis: Nifty 50 risk premium (range: -0.015 to 0.015)
- Y-axis: Reliance risk premium (range: -0.03 to 0.03)

The plot features purple data points, a blue regression line, and a grey grid. Two horizontal and vertical skyblue lines are drawn at zero on both axes. An NPTEL logo is visible in the top right corner of the RStudio window. A small inset image of a person is visible in the bottom right corner of the slide.

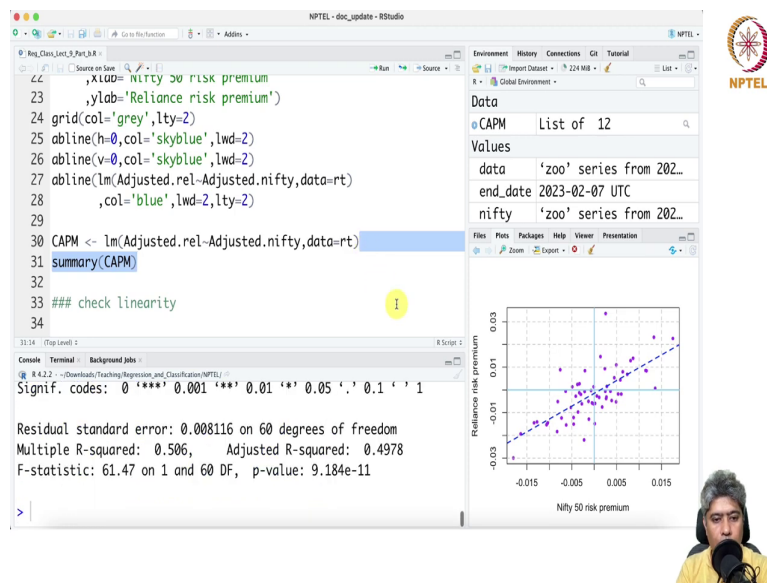
And then what it does that it fit the Reliance Industry as a function of nifty and then it just yeah.

(Refer Slide Time: 10:16)



So, it just create this line through the almost through the center.

(Refer Slide Time: 10:25)



The screenshot displays the RStudio interface with the following elements:

- Source Editor:** Contains R code for plotting and fitting a CAPM model. The code includes:

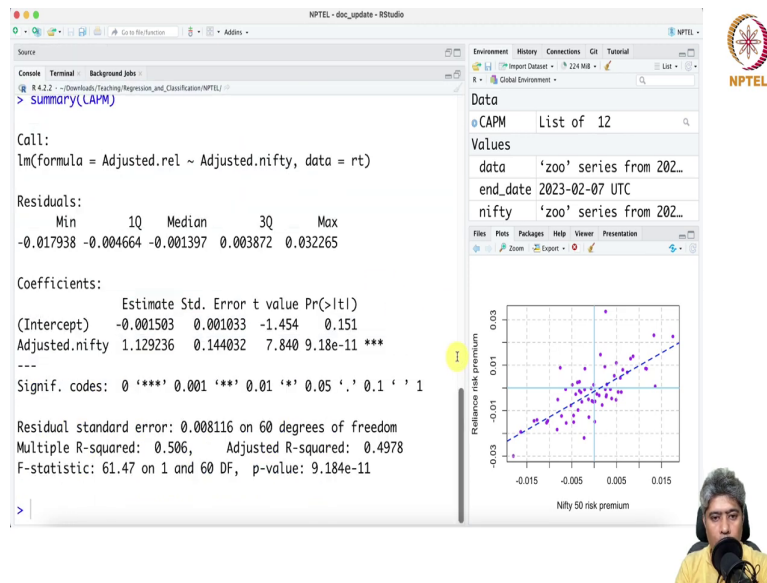
```
22 x.LGD = nifty ~ risk premium
23 ,ylab='Reliance risk premium')
24 grid(col='grey',lty=2)
25 abline(h=0,col='skyblue',lwd=2)
26 abline(v=0,col='skyblue',lwd=2)
27 abline(lm(Adjusted.rel-Adjusted.nifty,data=rt)
28 ,col='blue',lwd=2,lty=2)
29
30 CAPM <- lm(Adjusted.rel-Adjusted.nifty,data=rt)
31 summary(CAPM)
32
33 ## check linearity
34
```
- Environment:** Shows 'CAPM' as a 'List of 12'.
- Values:** Lists 'data' as 'zoo' series from 202..., 'end_date' as 2023-02-07 UTC, and 'nifty' as 'zoo' series from 202....
- Console:** Displays the output of the CAPM model:

```
Stgnif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.008116 on 60 degrees of freedom
Multiple R-squared: 0.506, Adjusted R-squared: 0.4978
F-statistic: 61.47 on 1 and 60 DF, p-value: 9.184e-11
```
- Plots:** A scatter plot titled 'Reliance risk premium' vs 'Nifty 50 risk premium'. The plot shows a positive linear relationship with a blue regression line and a grey grid. The axes range from -0.03 to 0.03.
- NPTEL Logo:** Located in the top right corner.
- Speaker:** A small video feed of a person speaking is visible in the bottom right corner.

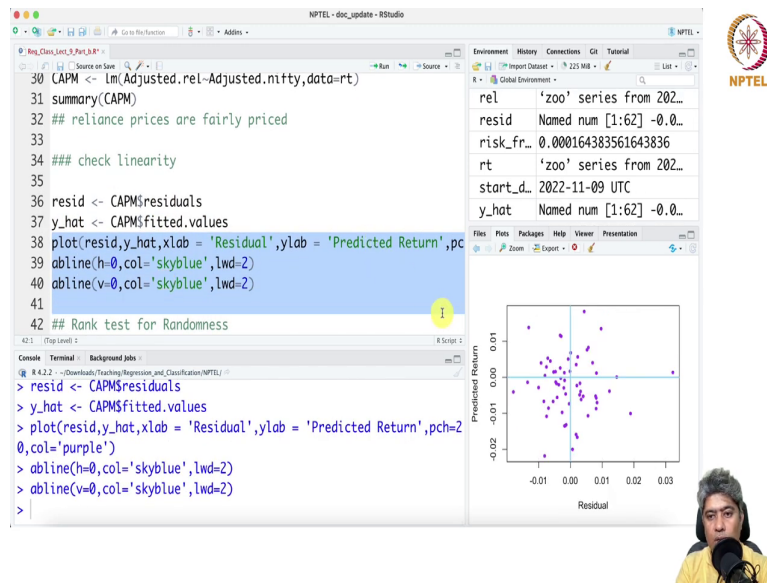
So, now if I fit CAPM just all you have to do Reliance the risk premium of Reliance as the response and risk premium of Nifty market index as the predictor call lm and provide the data-rt rt contains all the risk premium after subtracting the risk-free rate from the log return. So, now you run the CAPM and run the summary.

(Refer Slide Time: 10:59)



So, this is the alpha and this is the beta. So, beta is 1.12 and alpha is negative 0.00153, but P value is 0.151. So, we cannot really reject the null hypothesis that alpha is non-zero.

(Refer Slide Time: 11:26)



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

```
30 CAPM <- lm(Adjusted.rel~Adjusted.nitty,data=rt)
31 summary(CAPM)
32 ## reliance prices are fairly priced
33
34 ### check linearity
35
36 resid <- CAPM$residuals
37 y_hat <- CAPM$fitted.values
38 plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pch=2)
39 abline(h=0,col='skyblue',lwd=2)
40 abline(v=0,col='skyblue',lwd=2)
41
42 ## Rank test for Randomness
```

The console shows the execution of the code:

```
> resid <- CAPM$residuals
> y_hat <- CAPM$fitted.values
> plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pch=2,col='purple')
> abline(h=0,col='skyblue',lwd=2)
> abline(v=0,col='skyblue',lwd=2)
>
```

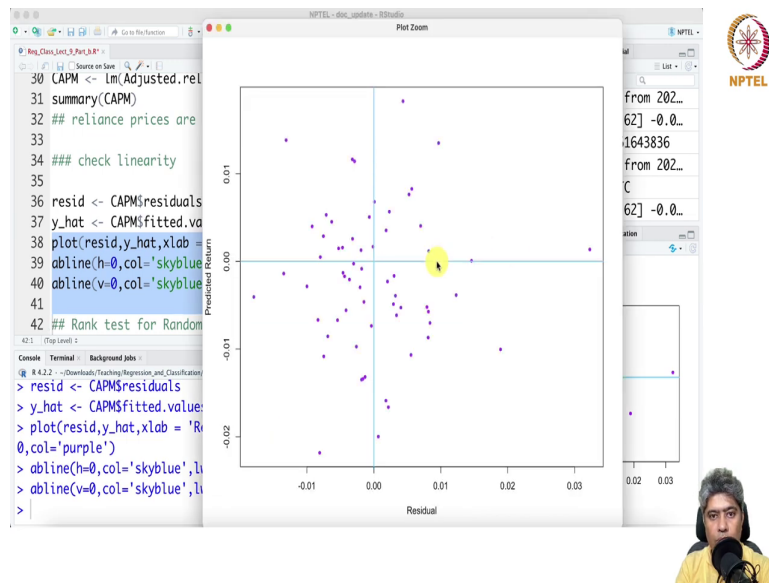
The Environment pane on the right shows the following objects:

rel	'zoo' series from 202...
resid	Named num [1:62] -0.0...
risk_fr...	0.000164383561643836
rt	'zoo' series from 202...
start_d...	2022-11-09 UTC
y_hat	Named num [1:62] -0.0...

The Plots pane shows a scatter plot of Predicted Return (y-axis) versus Residual (x-axis). The data points are purple dots scattered around the origin. Two thick skyblue lines are drawn: a horizontal line at y=0 and a vertical line at x=0. The NPTEL logo is visible in the top right corner of the RStudio window.

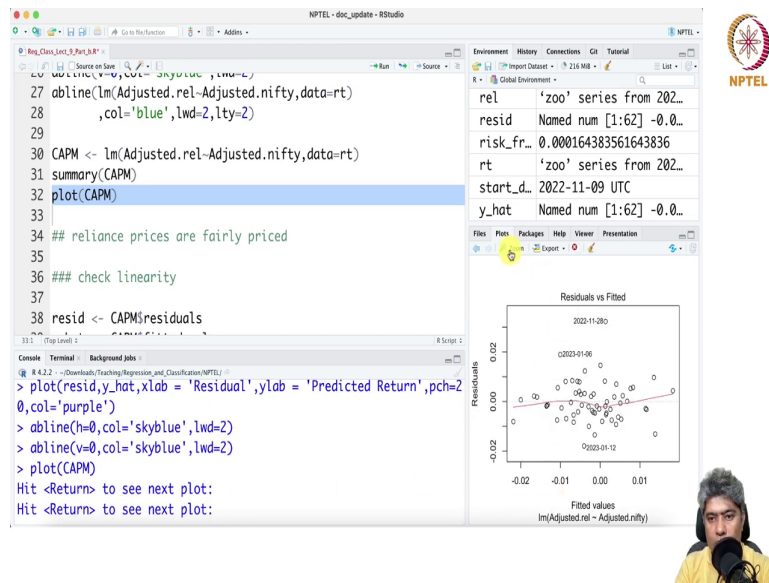
So, we can say that reliance prices are fairly priced. Ok. First we would like to do some quick linearity check. So, first we will take the residuals, take the y hat and do some plot of residual versus y hat.

(Refer Slide Time: 11:57)



So, here is the residual and here is the y hat. You can do one more thing also.

(Refer Slide Time: 12:06)



The screenshot displays the RStudio interface with the following components:

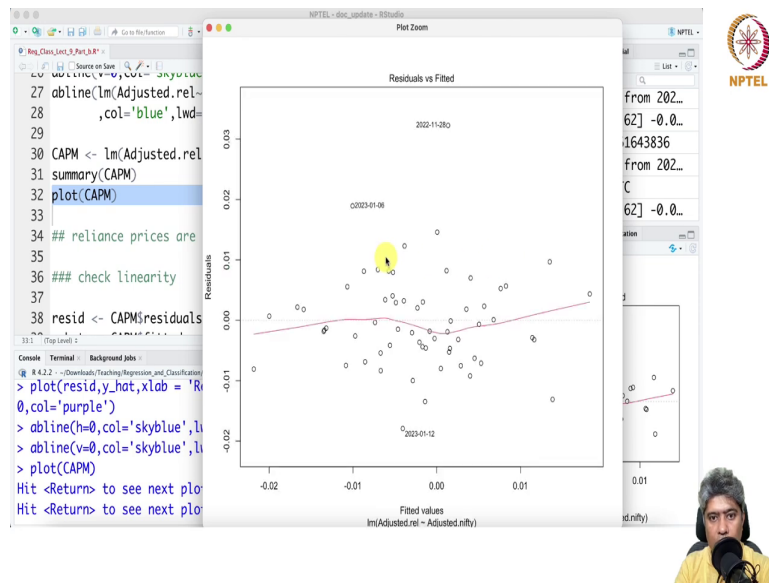
- Source Editor:** Contains R code for fitting a CAPM model and plotting residuals. The code includes:

```
27 abline(lm(Adjusted.rel-Adjusted.nifty,data=rt)
28       ,col='blue',lwd=2,lty=2)
29
30 CAPM <- lm(Adjusted.rel-Adjusted.nifty,data=rt)
31 summary(CAPM)
32 plot(CAPM)
33
34 ## reliance prices are fairly priced
35
36 ### check linearity
37
38 resid <- CAPM$residuals
```
- Environment:** Lists variables: rel ('zoo' series from 202...), resid (Named num [1:62] -0.0...), risk_fr... (0.000164383561643836), rt ('zoo' series from 202...), start_d... (2022-11-09 UTC), and y_hat (Named num [1:62] -0.0...).
- Console:** Shows the execution of the plot command and the resulting output:

```
> plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pch=2
0,col='purple')
> abline(h=0,col='skyblue',lwd=2)
> abline(v=0,col='skyblue',lwd=2)
> plot(CAPM)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```
- Plots:** A 'Residuals vs Fitted' plot is shown. The y-axis is labeled 'Residuals' and the x-axis is 'Fitted values lm(Adjusted.rel - Adjusted.nifty)'. The plot shows a scatter of points around a horizontal line at zero, with a vertical line at zero. The plot is titled '2022-11-28'.

So, now here you have fitted plot if you just fit plot CAPM if you just fit plot CAPM you will see a bunch of plot will get created. This is a built-in function in R.

(Refer Slide Time: 12:21)



So, this is the first plot that you will get. So, here you have fitted value versus residual ok. Fitted value versus residual and some of the points are like sort of an outlier.

(Refer Slide Time: 12:35)

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

```
27 abline(lm(Adjusted.rel-Adjusted.nifty,data=rt)
28       ,col='blue',lwd=2,pty=2)
29
30 CAPM <- lm(Adjusted.rel-Adjusted.nifty,data=rt)
31 summary(CAPM)
32 plot(CAPM)
33
34 ## reliance prices are fairly priced
35
36 ### check linearity
37
38 resid <- CAPM$residuals
```

The console shows the execution of the following commands:

```
0,col='purple')
> abline(h=0,col='skyblue',lwd=2)
> abline(v=0,col='skyblue',lwd=2)
> plot(CAPM)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

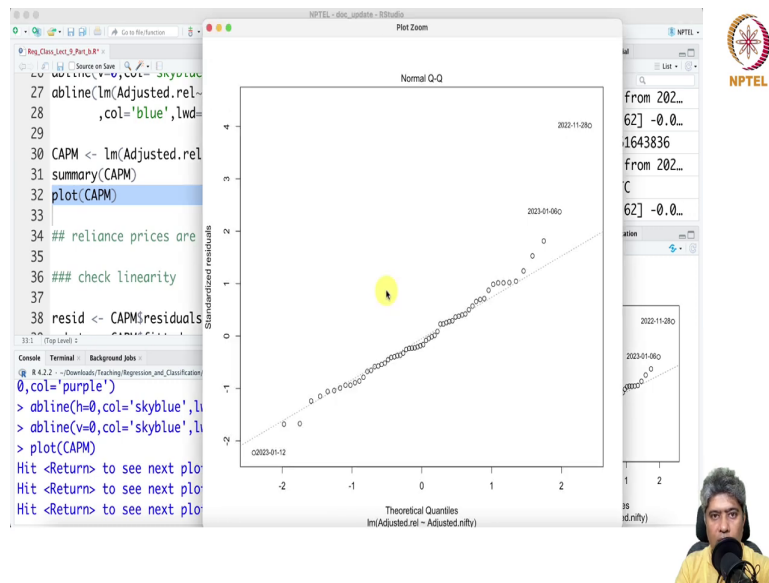
The environment pane on the right shows the following variables:

rel	'zoo' series from 202...
resid	Named num [1:62] -0.0...
risk_fr...	0.000164383561643836
rt	'zoo' series from 202...
start_d...	2022-11-09 UTC
y_hat	Named num [1:62] -0.0...

The plot pane shows a Normal Q-Q plot of the standardized residuals. The x-axis is labeled 'Theoretical Quantiles' and the y-axis is 'Standardized Residuals'. The data points follow a diagonal line, indicating a normal distribution. Two vertical blue lines are drawn at x=0 and x=1. Two horizontal blue lines are drawn at y=0 and y=1. The plot is titled 'Normal Q-Q' and includes the text 'lm(Adjusted.rel - Adjusted.nifty)'.

But the and then second plot it will be QQ plot.

(Refer Slide Time: 12:38)



It says that looks like some of the points are you know it is bit away from the like QQ line normality is looks like questionable.

(Refer Slide Time: 12:49)

The screenshot displays the RStudio interface with the following R code in the script editor:

```
27 abline(lm(Adjusted.rel-Adjusted.nifty,data=rt)
28       ,col='blue',lwd=2,lty=2)
30 CAPM <- lm(Adjusted.rel-Adjusted.nifty,data=rt)
31 summary(CAPM)
32 plot(CAPM)
33
34 ## reliance prices are fairly priced
35
36 ### check linearity
37
38 resid <- CAPM$residuals
```



The console shows the execution of the following commands:

```
> abline(h=0,col='skyblue',lwd=2)
> abline(v=0,col='skyblue',lwd=2)
> plot(CAPM)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

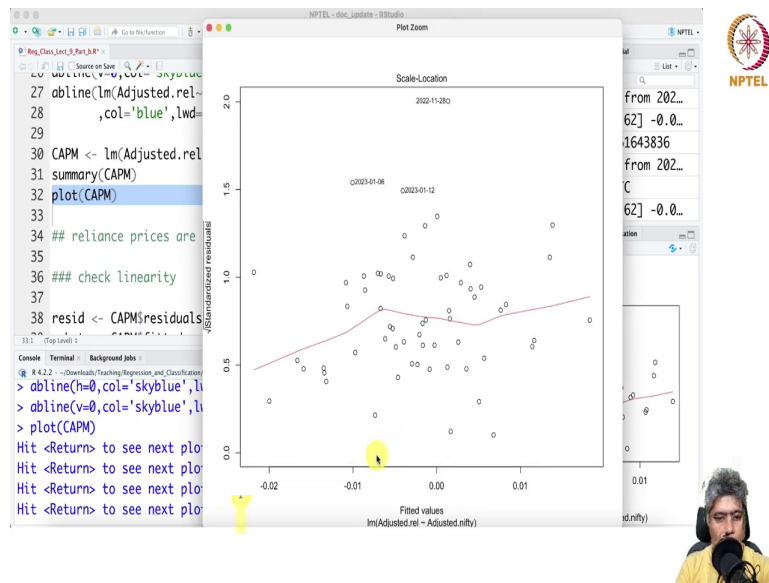
The environment pane on the right lists the following objects:

- rel: 'zoo' series from 202...
- resid: Named num [1:62] -0.0...
- risk_fr...: 0.000164383561643836
- rt: 'zoo' series from 202...
- start_d...: 2022-11-09 UTC
- y_hat: Named num [1:62] -0.0...

The plot pane shows a scatter plot titled "Scale-Location" for the date "2022-11-26". The y-axis is labeled "-Standardized residuals" and ranges from -2.0 to 2.0. The x-axis is labeled "Fitted values lm(Adjusted.rel - Adjusted.nifty)" and ranges from -0.02 to 0.01. A red regression line is plotted through the data points. A yellow circle with the number '1' is visible in the bottom right corner of the plot area.



(Refer Slide Time: 12:52)



Then third is a fitted values versus standardized residual and there are some huge outlier as usual we were seeing and this is the this is the leverage versus standardized residuals.

(Refer Slide Time: 13:01)

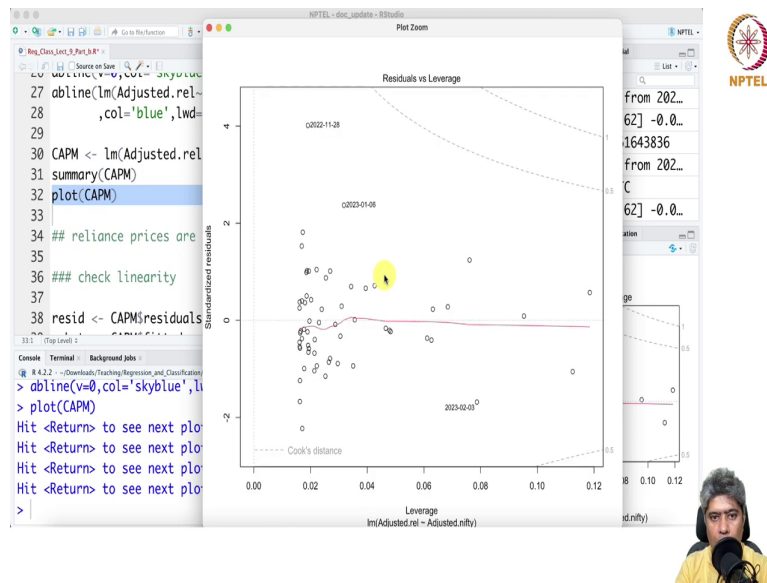
The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for fitting a linear model and generating diagnostic plots.

```
27 abline(lm(Adjusted.rel~Adjusted.nifty,data=rt),  
28         col='blue',lwd=2,lty=2)  
30 CAPM <- lm(Adjusted.rel~Adjusted.nifty,data=rt)  
31 summary(CAPM)  
32 plot(CAPM)  
33  
34 ## reliance prices are fairly priced  
35  
36 ### check linearity  
37  
38 resid <- CAPM$residuals
```
- Environment:** Lists variables in the workspace: `rel` (series from 202...), `resid` (Named num [1:62] -0.0...), `risk_fr...` (0.000164383561643836), `rt` (series from 202...), `start_d...` (2022-11-09 UTC), and `y_hat` (Named num [1:62] -0.0...).
- Console:** Shows the execution of `abline` and `plot` commands, with prompts for the next plot.

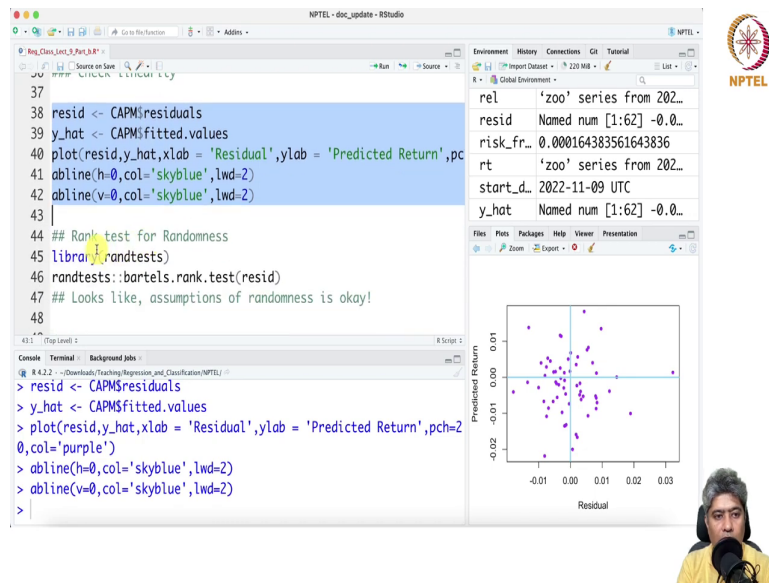
```
> abline(v=0,col='skyblue',lwd=2)  
> plot(CAPM)  
Hit <Return> to see next plot:  
Hit <Return> to see next plot:  
Hit <Return> to see next plot:  
Hit <Return> to see next plot:  
>
```
- Plots:** A 'Residuals vs Leverage' plot is shown. The y-axis is 'Standardized residuals' and the x-axis is 'Leverage lm(Adjusted.rel ~ Adjusted.nifty)'. A horizontal blue line is at y=0. Two points are highlighted with dashed lines: '2022-11-09' (high leverage, positive residual) and '2022-02-09' (high leverage, negative residual). A yellow circle with the letter 'I' is placed near the console output.
- Background:** The NPTEL logo is visible in the top right corner, and a small video feed of a person is in the bottom right corner.

(Refer Slide Time: 13:04)



I have not discussed what is leverage and cooks distance all those things going forward. I will this I will teach you what is leverage what is cooks distance all those issues. So, at this moment I will not be able to explain this particular plot, but I am hoping by in next two weeks, I will be able to teach you what is leverage and what is residuals and at that time I will be able to explain this plot to you.

(Refer Slide Time: 13:41)





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

```
37  
38 resid <- CAPM$residuals  
39 y_hat <- CAPM$fitted.values  
40 plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pch=2  
41 abline(h=0,col='skyblue',lwd=2)  
42 abline(v=0,col='skyblue',lwd=2)  
43 |  
44 ## Rank test for Randomness  
45 library(randtests)  
46 randtests::bartels.rank.test(resid)  
47 ## Looks like, assumptions of randomness is okay!  
48
```

The console on the right shows the execution of the code:

```
> resid <- CAPM$residuals  
> y_hat <- CAPM$fitted.values  
> plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pch=2  
0,col='purple')  
> abline(h=0,col='skyblue',lwd=2)  
> abline(v=0,col='skyblue',lwd=2)  
>
```

The environment pane on the right lists variables: rel, resid, risk_fr, rt, start_d, and y_hat. The plot window shows a scatter plot of Predicted Return (y-axis, ranging from -0.02 to 0.01) versus Residual (x-axis, ranging from -0.01 to 0.03). The data points are purple circles. Two horizontal and two vertical skyblue lines are overlaid on the plot, representing the zero lines for both axes.



So, our usual things I will check. So, here we have done a quick thing as of then we have test for randomness test.

(Refer Slide Time: 13:54)



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

```
37  
38 resid <- CAPM$residuals  
39 y_hat <- CAPM$fitted.values  
40 plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pc  
41 abline(h=0,col='skyblue',lwd=2)  
42 abline(v=0,col='skyblue',lwd=2)  
43  
44 ## Rank test for Randomness  
45 library(rantests)  
46 rantests::bartels.rank.test(resid)  
47 ## Looks like, assumptions of randomness is okay!  
48
```

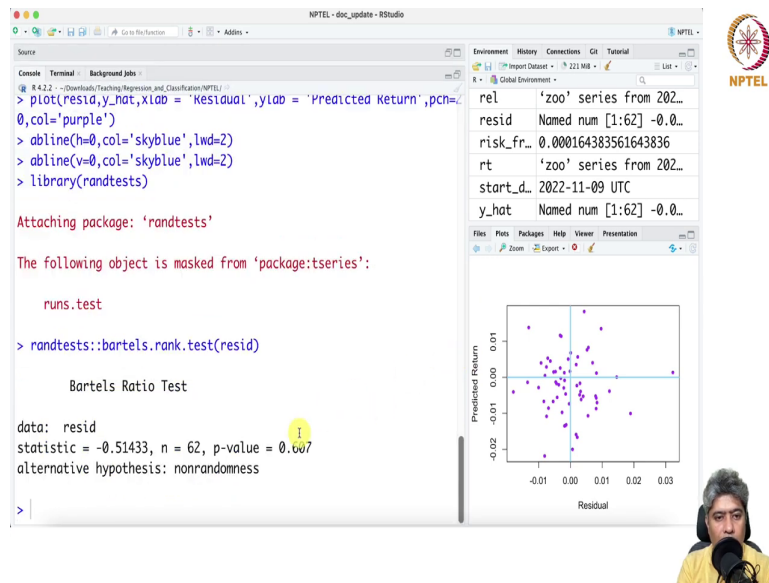
The console output shows the results of the Bartels Ratio Test:

```
Bartels Ratio Test  
  
data: resid  
statistic = -0.51433, n = 62, p-value = 0.607  
alternative hypothesis: nonrandomness
```

The environment pane on the right lists variables: rel, resid, risk_fr, rt, start_d, and y_hat. The plot pane shows a scatter plot of Predicted Return (y-axis) versus Residual (x-axis). The plot includes two horizontal blue lines at y=0 and two vertical blue lines at x=0, forming a cross. The data points are purple dots scattered around the origin.



(Refer Slide Time: 13:56)



The screenshot shows the RStudio interface. The console window displays the following R code and output:

```
> plot(resid, y_hat, xlab = 'Residual', ylab = 'Predicted Return', pch=2, col='purple')
> abline(h=0, col='skyblue', lwd=2)
> abline(v=0, col='skyblue', lwd=2)
> library(randtests)

Attaching package: 'randtests'

The following object is masked from 'package:tseries':

    runs.test

> randtests::bartels.rank.test(resid)

Bartels Ratio Test

data: resid
statistic = -0.51433, n = 62, p-value = 0.607
alternative hypothesis: nonrandomness
```

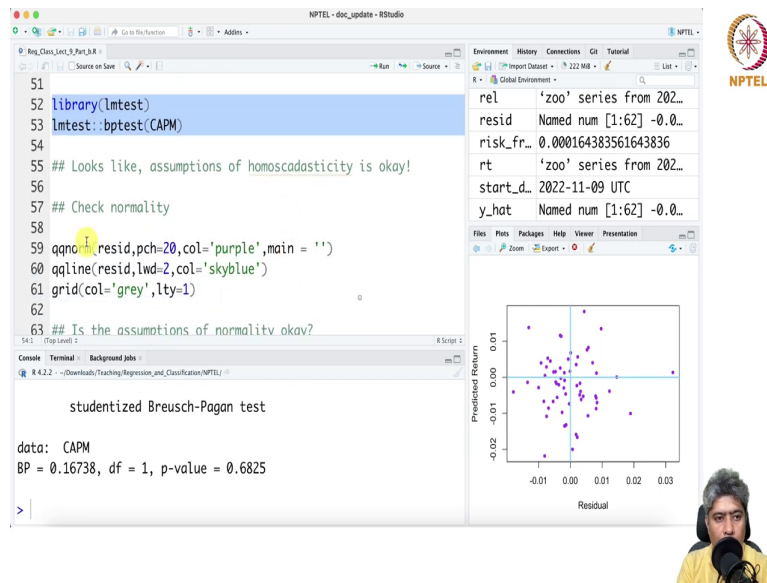
The environment pane on the right shows the following objects:

Object	Class	Value
rel	'zoo' series from 202...	
resid	Named num [1:62]	-0.0...
risk_fr...	0.000164383561643836	
rt	'zoo' series from 202...	
start_d...	2022-11-09 UTC	
y_hat	Named num [1:62]	-0.0...

The plot window shows a scatter plot of Predicted Return (y-axis) versus Residual (x-axis). The data points are purple dots scattered around the origin. Two blue lines are drawn: a horizontal line at y=0 and a vertical line at x=0. The NPTEL logo is visible in the top right corner of the RStudio window. A small video inset of a person speaking is visible in the bottom right corner of the RStudio window.

And what we found that p value is bit high. So, randomness is ok. So, the assumption of randomness is ok then we can run the Breusch Pagan Test for homoscedasticity.

(Refer Slide Time: 14:11)




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


```
51  
52 library(lmtest)  
53 lmtest::bptest(CAPM)  
54  
55 ## Looks like, assumptions of homoscedasticity is okay!  
56  
57 ## Check normality  
58  
59 qqnorm(resid,pch=20,col='purple',main = '')  
60 qqline(resid,lwd=2,col='skyblue')  
61 grid(col='grey',lty=1)  
62  
63 ## Is the assumptions of normality okay?
```

The console on the right shows the output of the Breusch-Pagan test:

```
studentized Breusch-Pagan test  
  
data: CAPM  
BP = 0.16738, df = 1, p-value = 0.6825
```

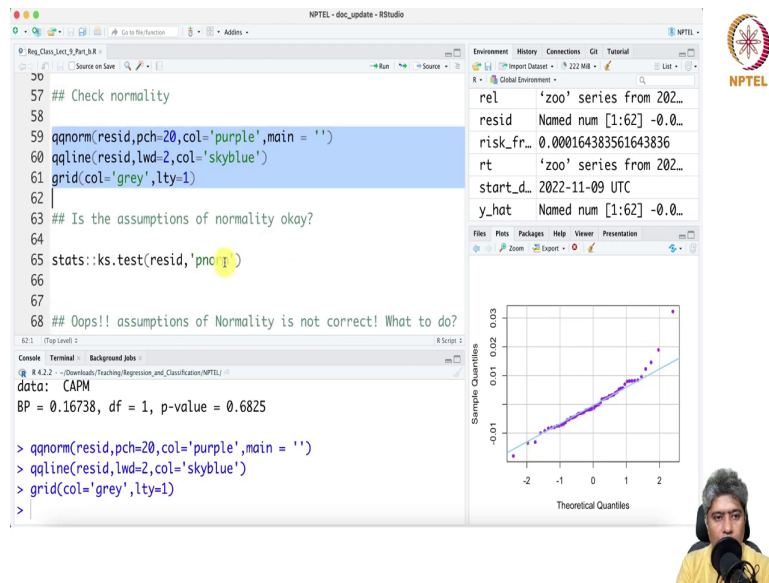
The environment pane on the right lists variables: rel, resid, risk_fr, rt, start_d, and y_hat. The plot pane shows a residual plot with purple points and a blue horizontal line at zero. The x-axis is labeled 'Residual' and the y-axis is labeled 'Predicted Return'.





So, let us run the Breusch Pagan test for homoscedasticity and p-value we can see p-value for Breusch Pagan test is also very large. So, the assumption for homoscedasticity is also ok. Now, you run check for normality. So, looks like these points are far away from the QQ line. So, there is a questionable thing is the assumption of normality ok.

(Refer Slide Time: 14:24)



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

```
57 ## Check normality
58
59 qqnorm(resid,pch=20,col='purple',main = '')
60 qqline(resid,lwd=2,col='skyblue')
61 grid(col='grey',lty=1)
62
63 ## Is the assumptions of normality okay?
64
65 stats::ks.test(resid,'pnorm')
66
67
68 ## Oops!! assumptions of Normality is not correct! What to do?
```

The console on the bottom left shows the output of the `ks.test` function:

```
data: CAPM
BP = 0.16738, df = 1, p-value = 0.6825

> qqnorm(resid,pch=20,col='purple',main = '')
> qqline(resid,lwd=2,col='skyblue')
> grid(col='grey',lty=1)
>
```

The environment pane on the right lists variables: `rel` ('zoo' series from 202...), `resid` (Named num [1:62] -0.0...), `risk_fr...` (0.000164383561643836), `rt` ('zoo' series from 202...), `start_d...` (2022-11-09 UTC), and `y_hat` (Named num [1:62] -0.0...).

The plot pane on the right shows a Q-Q plot with 'Sample Quantiles' on the y-axis and 'Theoretical Quantiles' on the x-axis. The data points are purple circles, and a blue line represents the theoretical normal distribution. The points deviate from the line, indicating non-normality.

The NPTEL logo is visible in the top right corner, and a small video feed of a person is in the bottom right corner.

(Refer Slide Time: 14:35)

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

```
60 qqline(resid,lwd=2,col='skyblue')
61 grid(col='grey',lty=1)
62
63 ## Is the assumptions of normality okay?
64
65 stats::ks.test(resid,'pnorm')
66
67
68 ## Oops!! assumptions of Normality is not correct! What to do?
69
```

The console output shows the result of the test:

```
> stats::ks.test(resid,'pnorm')
Exact one-sample Kolmogorov-Smirnov test
data: resid
D = 0.49284, p-value = 2.054e-14
alternative hypothesis: two-sided
```

The environment pane on the right shows the following objects:

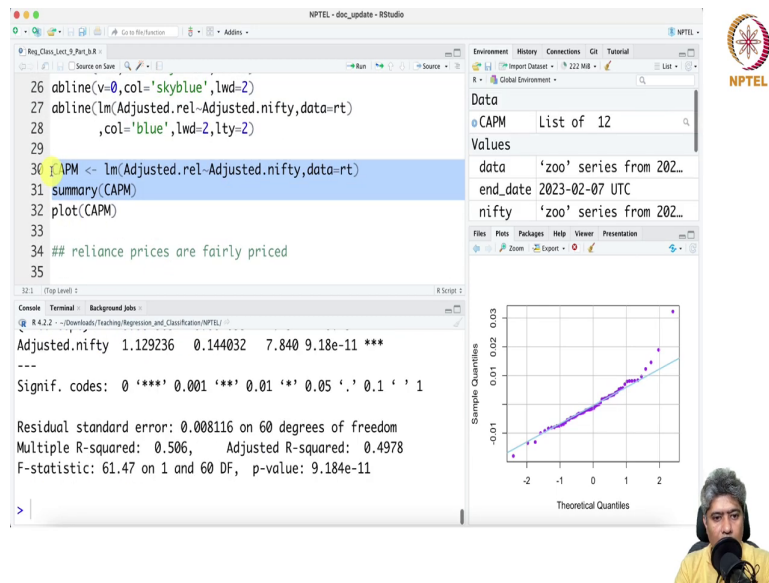
Object	Class	Attributes
rel	'zoo' series from 202...	
resid	Named num [1:62]	-0.0...
risk_fr...	0.000164383561643836	
rt	'zoo' series from 202...	
start_d...	2022-11-09 UTC	
y_hat	Named num [1:62]	-0.0...

The plot pane shows a Q-Q plot with 'Sample Quantiles' on the y-axis and 'Theoretical Quantiles' on the x-axis. The data points are purple dots, and a blue line represents the theoretical normal distribution. The points deviate significantly from the line, indicating non-normality.

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

So, let us run the Kolmogorov Smirnov test and clearly p value is very very small. Very very small like after 0.14 zeros and then after 14 decimals place you are getting 2054. So, it is impossible. So, assumption of check normality is not correct. Therefore, this inference that we did if we this inference that we did.

(Refer Slide Time: 15:06)



The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for fitting a linear model and generating a CAPM plot.

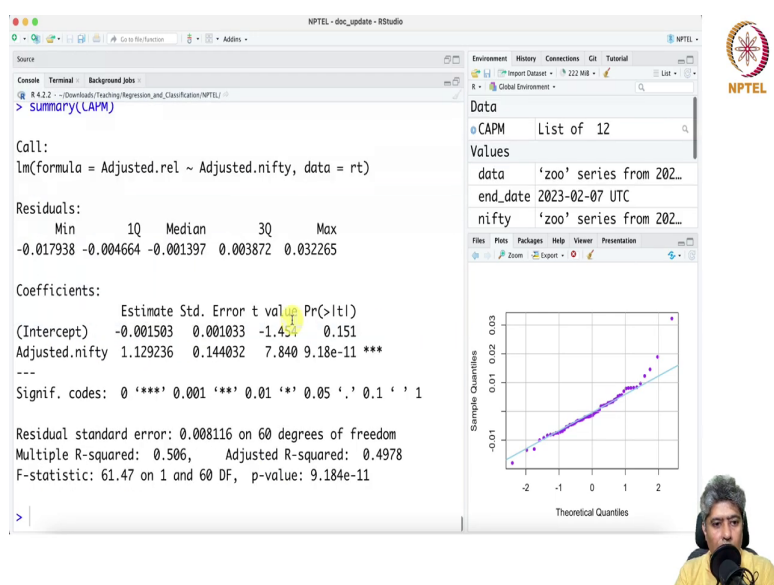
```
26 abline(v=0,col='skyblue',lwd=2)
27 abline(lm(Adjusted.rel-Adjusted.nifty,data=rt)
28       ,col='blue',lwd=2,pty=2)
29
30 CAPM <- lm(Adjusted.rel-Adjusted.nifty,data=rt)
31 summary(CAPM)
32 plot(CAPM)
33
34 ## reliance prices are fairly priced
35
```
- Environment:** Shows the 'Data' environment with variables 'CAPM' (List of 12) and 'Values' (data, end_date, nifty).
- Console:** Displays the output of the `summary(CAPM)` command:

```
Adjusted.nifty 1.129236 0.144032 7.840 9.18e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.008116 on 60 degrees of freedom
Multiple R-squared:  0.506,    Adjusted R-squared:  0.4978
F-statistic: 61.47 on 1 and 60 DF,  p-value: 9.184e-11
```
- Plots:** A Q-Q plot showing 'Sample Quantiles' on the y-axis (ranging from -0.01 to 0.03) and 'Theoretical Quantiles' on the x-axis (ranging from -2 to 2). The data points closely follow a diagonal reference line, indicating a normal distribution.

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

(Refer Slide Time: 15:08)



This inference is now not anymore valid because this test is done based on t value and t value requires the assumption of normality. And clearly our residual does not follow normality. So, this assumption these entire analysis is not valid anymore ok. So, in order to solve this situation what typically we do? One possible solution is bootstrap regression and non-parametric bootstrap regression.

So, at this moment I am going to stop here and in the next video I will start discussing what is bootstrap regression and then I will do the hands on how we can solve this problem of capital asset pricing model using bootstrap regression. I hope you enjoyed this video.

Thank you very much. See you in the next video.

