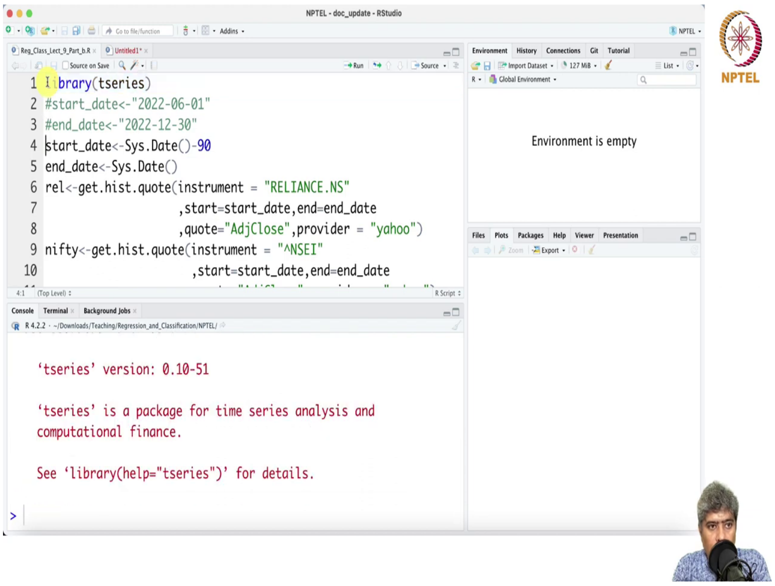
Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Chennai Mathematical Institute

Lecture - 33
Hands on with R for Bootstrap Regression

Hello all welcome back to the last part or the final part of lecture 9. In this video, we will be doing some hands on, on how to do Bootstrap Regression for capital asset pricing model.

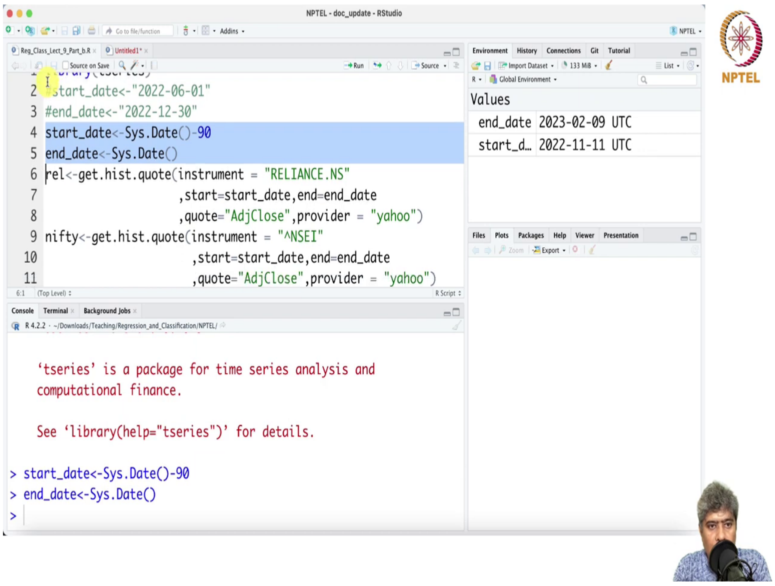
(Refer Slide Time: 00:31)



```
library(tseries)
#start_date<-"2022-06-01"
#end_date<-"2022-12-30"
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")
nifty<-get.hist.quote(instrument = "^NSEI"
                      ,start=start_date,end=end_date
```

```
'tseries' version: 0.10-51

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

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

>
```

(Refer Slide Time: 00:48)



```
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")
```

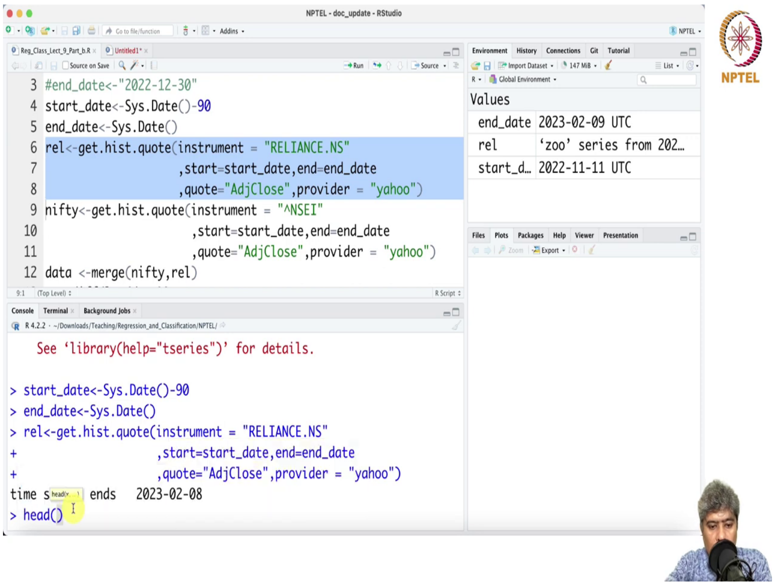
Values

end_date	2023-02-09 UTC
start_d…	2022-11-11 UTC

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

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

> start_date<-Sys.Date()-90
> end_date<-Sys.Date()
> 
```

So, here in our this piece of code you have already seen, I will quickly remind you that you know we are going to use a T series package, then we are going to use system date or start date. Here we are then now this piece of code 9678 is going to download the data from internet from Yahoo Finance.

(Refer Slide Time: 01:01)



```r
#end_date<-"2022-12-30"
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")
nifty<-get.hist.quote(instrument = "^NSEI"
                      ,start=start_date,end=end_date
                      ,quote="AdjClose",provider = "yahoo")
data <-merge(nifty,rel)
```

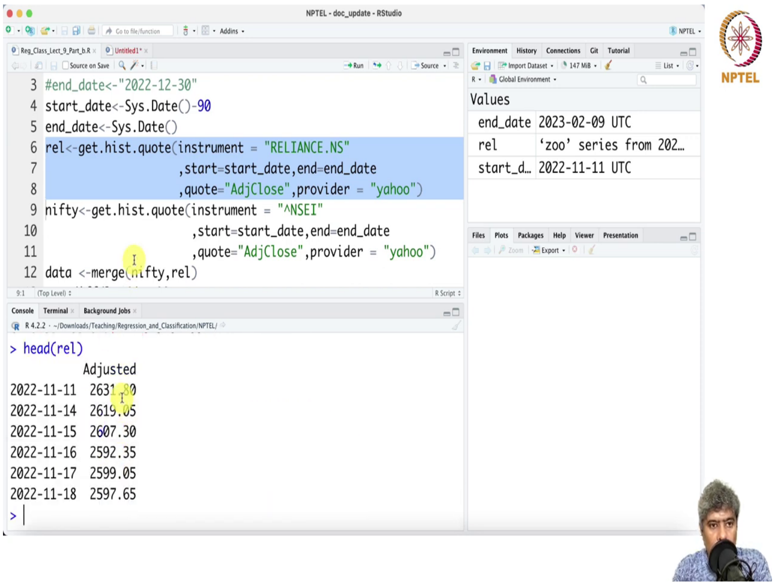
Values

end_date	2023-02-09 UTC
rel	‘zoo’ series from 202…
start_d…	2022-11-11 UTC

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

```
    See ‘library(help="tseries")’ for details.

> 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 s head(x, ...) ends   2023-02-08
> head()
```

(Refer Slide Time: 01:12)



```r
#end_date<-"2022-12-30"
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")
nifty<-get.hist.quote(instrument = "^NSEI"
                      ,start=start_date,end=end_date
                      ,quote="AdjClose",provider = "yahoo")
data <-merge(nifty,rel)
```

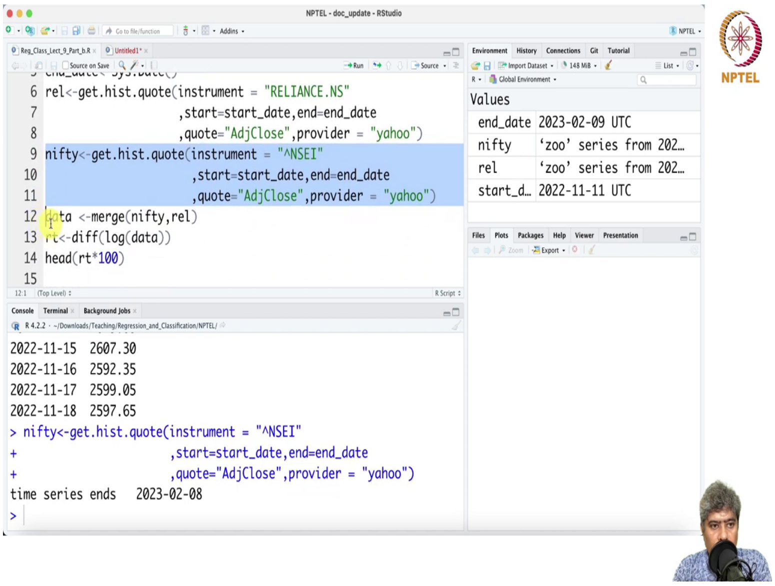
Values

end_date	2023-02-09 UTC
rel	'zoo' series from 202…
start_d…	2022-11-11 UTC

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

```
> head(rel)
           Adjusted
2022-11-11  2631.80
2022-11-14  2619.05
2022-11-15  2607.30
2022-11-16  2592.35
2022-11-17  2599.05
2022-11-18  2597.65
> 
```

(Refer Slide Time: 01:20)



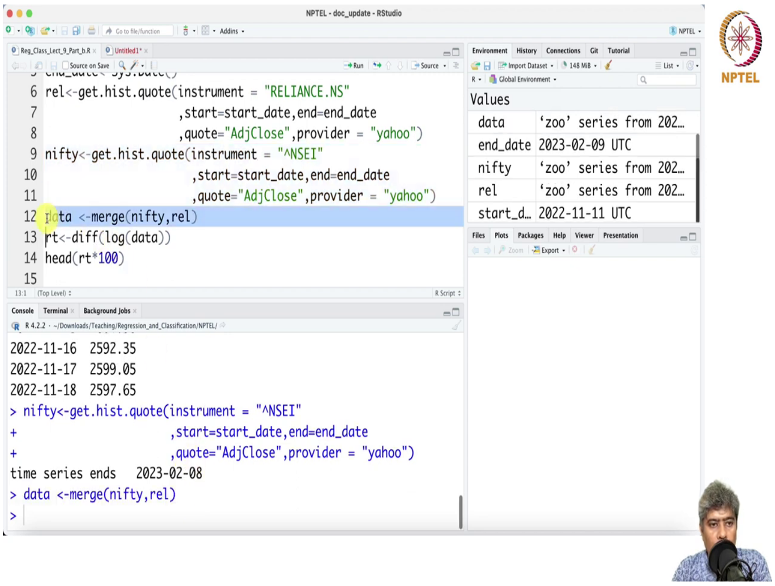
```
rel<-get.hist.quote(instrument = "RELIANCE.NS"
                    ,start=start_date,end=end_date
                    ,quote="AdjClose",provider = "yahoo")
nifty<-get.hist.quote(instrument = "^NSEI"
                      ,start=start_date,end=end_date
                      ,quote="AdjClose",provider = "yahoo")
data <-merge(nifty,rel)
rt<-diff(log(data))
head(rt*100)
```

```
2022-11-15  2607.30
2022-11-16  2592.35
2022-11-17  2599.05
2022-11-18  2597.65
> nifty<-get.hist.quote(instrument = "^NSEI"
+                       ,start=start_date,end=end_date
+                       ,quote="AdjClose",provider = "yahoo")
time series ends   2023-02-08
>
```

Values	
end_date	2023-02-09 UTC
nifty	'zoo' series from 202…
rel	'zoo' series from 202…
start_d…	2022-11-11 UTC

So, here we have data is downloaded from Yahoo Finance head rel. So, here we can see then NIFTY 50 data will be downloaded adjusted close prices as of 8th February 2023 all the data has come.

(Refer Slide Time: 01:28)



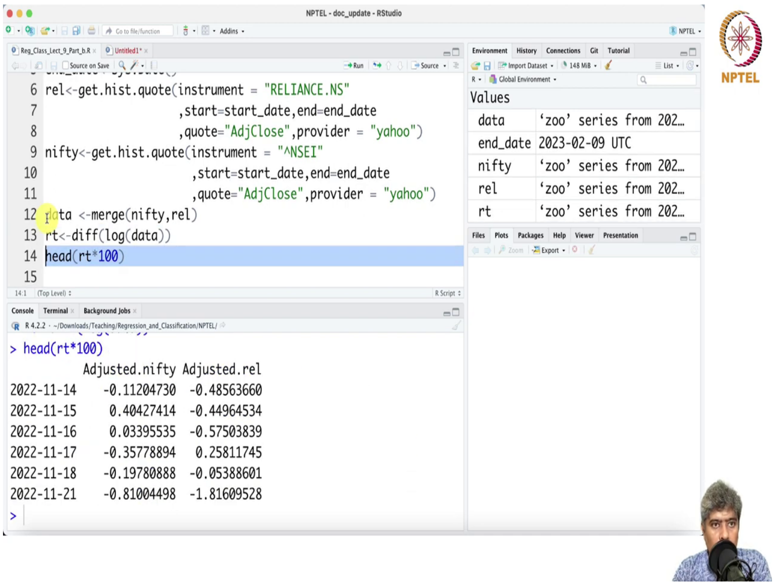
```
end_date<-Sys.Date()
rel<-get.hist.quote(instrument = "RELIANCE.NS"
                    ,start=start_date,end=end_date
                    ,quote="AdjClose",provider = "yahoo")
nifty<-get.hist.quote(instrument = "^NSEI"
                      ,start=start_date,end=end_date
                      ,quote="AdjClose",provider = "yahoo")
data <-merge(nifty,rel)
rt<-diff(log(data))
head(rt*100)
```

Values

data	‘zoo’ series from 202…
end_date	2023-02-09 UTC
nifty	‘zoo’ series from 202…
rel	‘zoo’ series from 202…
start_d…	2022-11-11 UTC

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
2022-11-16  2592.35
2022-11-17  2599.05
2022-11-18  2597.65
> nifty<-get.hist.quote(instrument = "^NSEI"
+                       ,start=start_date,end=end_date
+                       ,quote="AdjClose",provider = "yahoo")
time series ends   2023-02-08
> data <-merge(nifty,rel)
>
```

(Refer Slide Time: 01:37)



```
 6 rel<-get.hist.quote(instrument = "RELIANCE.NS"
                      ,start=start_date,end=end_date
                      ,quote="AdjClose",provider = "yahoo")
 9 nifty<-get.hist.quote(instrument = "^NSEI"
                       ,start=start_date,end=end_date
                       ,quote="AdjClose",provider = "yahoo")
12 data <-merge(nifty,rel)
13 rt<-diff(log(data))
14 head(rt*100)
15
```

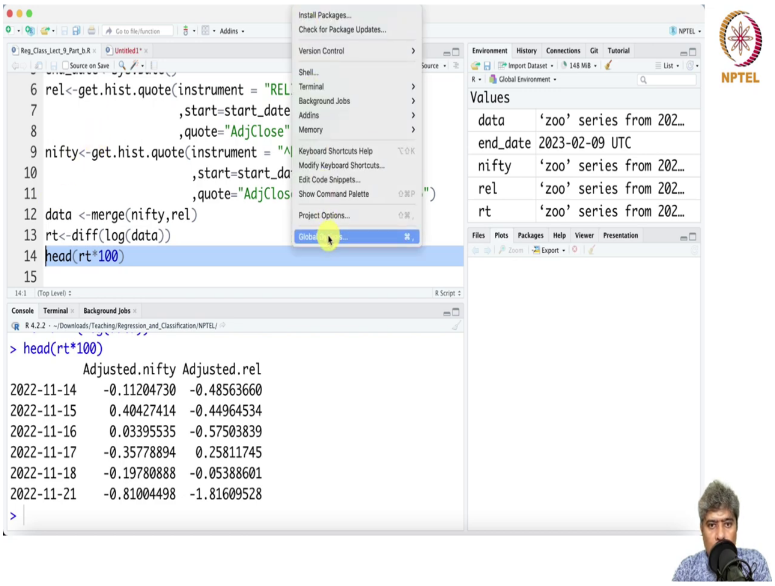
Values

data	‘zoo’ series from 202…
end_date	2023-02-09 UTC
nifty	‘zoo’ series from 202…
rel	‘zoo’ series from 202…
rt	‘zoo’ series from 202…

```
> head(rt*100)
           Adjusted.nifty Adjusted.rel
2022-11-14    -0.11204730  -0.48563660
2022-11-15     0.40427414  -0.44964534
2022-11-16     0.03395535  -0.57503839
2022-11-17    -0.35778894   0.25811745
2022-11-18    -0.19780888  -0.05388601
2022-11-21    -0.81004498  -1.81609528
>
```

Then we are going to merge these data here, in line 13 we are going to calculate the log return. Here is the log returns in percentage term.

(Refer Slide Time: 01:14)



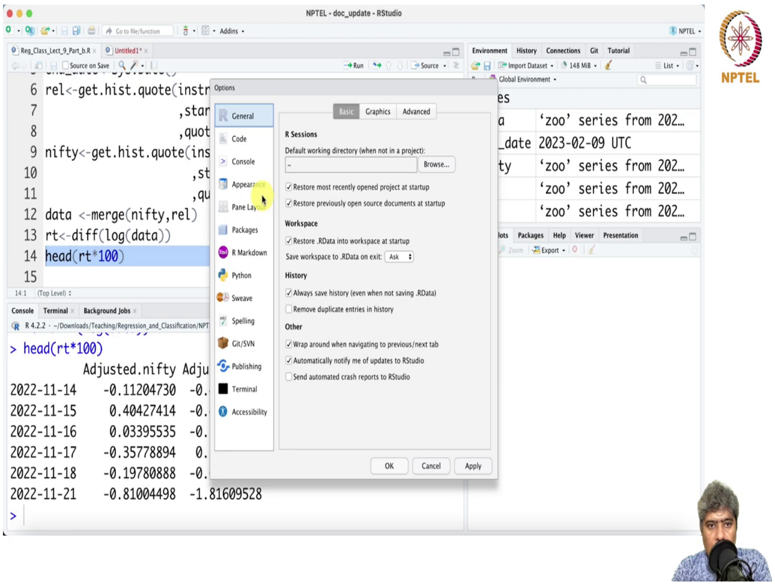
```
rel<-get.hist.quote(instrument = "REL
                    ,start=start_date
                    ,quote="AdjClose"
nifty<-get.hist.quote(instrument = "^
                      ,start=start_da
                      ,quote="AdjClos                ")
data <-merge(nifty,rel)
rt<-diff(log(data))
head(rt*100)
```

```
> head(rt*100)
           Adjusted.nifty Adjusted.rel
2022-11-14    -0.11204730  -0.48563660
2022-11-15     0.40427414  -0.44964534
2022-11-16     0.03395535  -0.57503839
2022-11-17    -0.35778894   0.25811745
2022-11-18    -0.19780888  -0.05388601
2022-11-21    -0.81004498  -1.81609528
>
```

Values

data	'zoo' series from 202…
end_date	2023-02-09 UTC
nifty	'zoo' series from 202…
rel	'zoo' series from 202…
rt	'zoo' series from 202…

(Refer Slide Time: 01:46)


NPTEL - doc_update - RStudio

```
rel<-get.hist.quote(instr
                  ,star
                  ,quot
nifty<-get.hist.quote(ins
                    ,st
                    ,qu
data <-merge(nifty,rel)
rt<-diff(log(data))
head(rt*100)
```

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPT

```
> head(rt*100)
           Adjusted.nifty Adju
2022-11-14    -0.11204730 -0.
2022-11-15     0.40427414 -0.
2022-11-16     0.03395535 -0.
2022-11-17    -0.35778894  0.
2022-11-18    -0.19780888 -0.
2022-11-21    -0.81004498 -1.81609528
>
```

```
es
a      'zoo' series from 202…
_date  2023-02-09 UTC
ty     'zoo' series from 202…
       'zoo' series from 202…
       'zoo' series from 202…
```

Options

General

Basic

R Sessions

Default working directory (when not in a project): ~

☑ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

Workspace

☑ Restore .RData into workspace at startup

Save workspace to .RData on exit: Ask

History

☑ Always save history (even when not saving .RData)

□ Remove duplicate entries in history

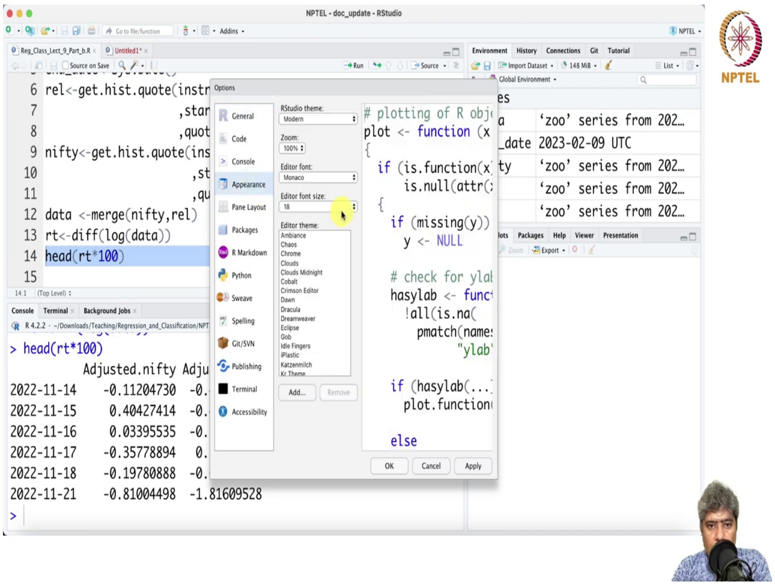
Other

☑ Wrap around when navigating to previous/next tab

☑ Automatically notify me of updates to RStudio

□ Send automated crash reports to RStudio

(Refer Slide Time: 01:47)



```
rel<-get.hist.quote(instr
                    ,star
                    ,quot
nifty<-get.hist.quote(ins
                      ,st
                      ,qu
data <-merge(nifty,rel)
rt<-diff(log(data))
head(rt*100)
```

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPT

```
> head(rt*100)
           Adjusted.nifty Adju
2022-11-14    -0.11204730  -0.
2022-11-15     0.40427414  -0.
2022-11-16     0.03395535  -0.
2022-11-17    -0.35778894   0.
2022-11-18    -0.19780888  -0.
2022-11-21    -0.81004498  -1.81609528
>
```

RStudio theme: Modern

Zoom: 100%

Editor font: Monaco

Editor font size: 18

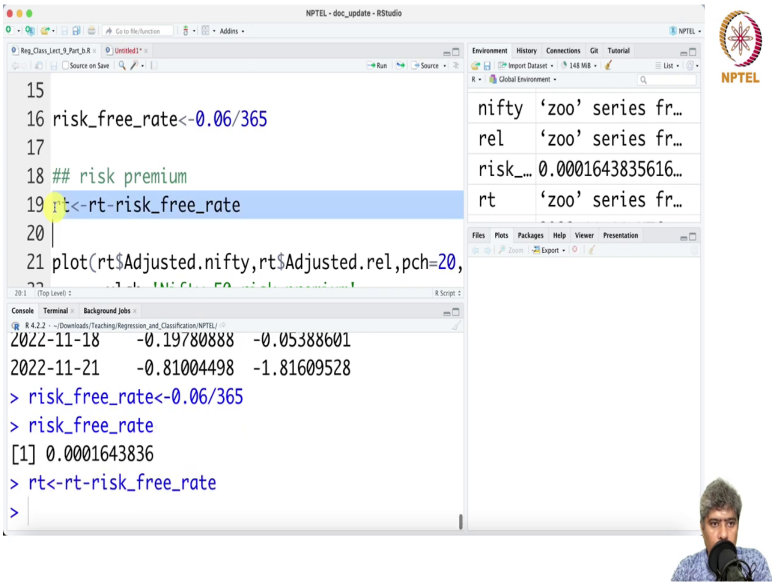
Editor theme:
Ambiance
Chaos
Chrome
Clouds
Clouds Midnight
Cobalt
Crimson Editor
Dawn
Dracula
Dreamweaver
Eclipse
Gob
Idle Fingers
iPlastic
Katzenmilch
Kr Theme

```
# plotting of R obje
plot <- function (x
{
  if (is.function(x)
      is.null(attr(
  {
    if (missing(y))
      y <- NULL

    # check for ylab
    hasylab <- funct
      !all(is.na(
        pmatch(names
               "ylab"

    if (hasylab(...)
      plot.function(

    else
```

```
a       'zoo' series from 202…
_date   2023-02-09 UTC
ty      'zoo' series from 202…
        'zoo' series from 202…
        'zoo' series from 202…
```

Let me increase the font size bit maybe I will use 24 apply ok; alright.

(Refer Slide Time: 01:58)



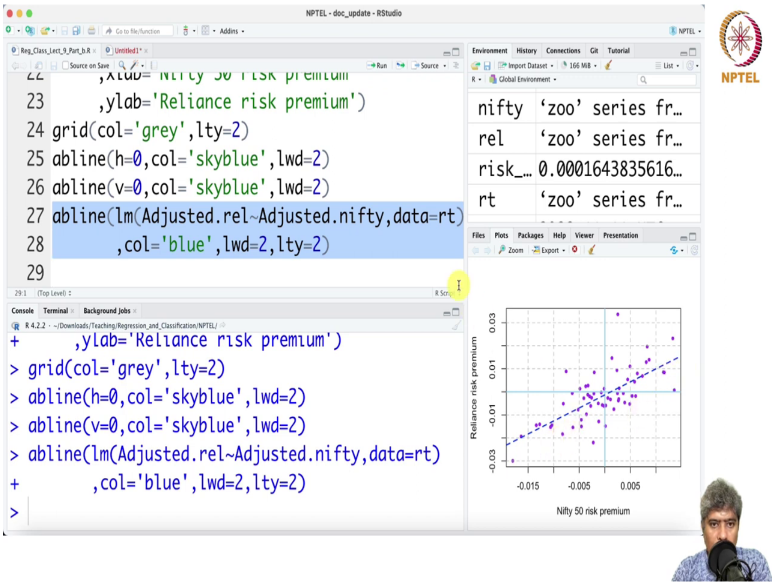
```
risk_free_rate<-0.06/365

## risk premium
rt<-rt-risk_free_rate

plot(rt$Adjusted.nifty,rt$Adjusted.rel,pch=20,
```

```
nifty   'zoo' series fr…
rel     'zoo' series fr…
risk_…  0.0001643835616…
rt      'zoo' series fr…
```

```
2022-11-18     -0.19780888  -0.05388601
2022-11-21     -0.81004498  -1.81609528
> risk_free_rate<-0.06/365
> risk_free_rate
[1] 0.0001643836
> rt<-rt-risk_free_rate
>
```

And now I am using risk free rate 6 percentage, annualized 6 percentages. So, I am dividing it to by 365, so this is the daily risk-free rate. Then I am calculating subtracting risk free rate from the return log return.

(Refer Slide Time: 02:24)



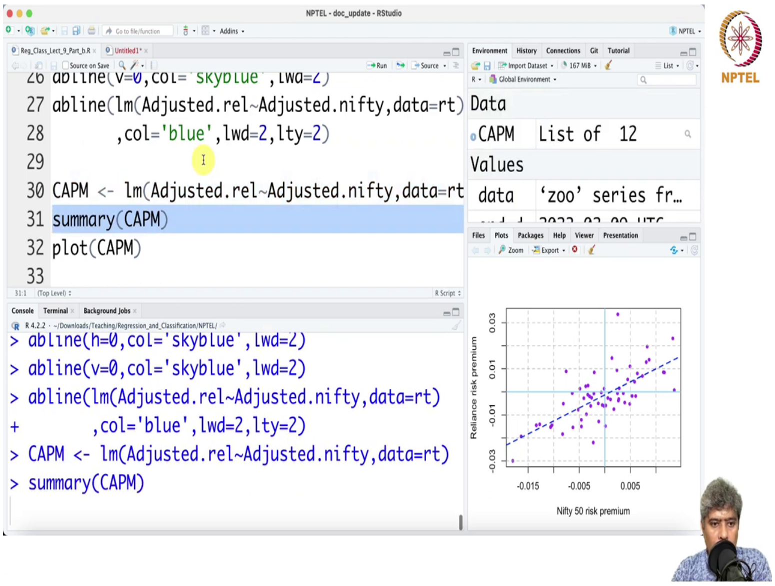
```
      ,xlab='Nifty 50 risk premium'
      ,ylab='Reliance risk premium')
grid(col='grey',lty=2)
abline(h=0,col='skyblue',lwd=2)
abline(v=0,col='skyblue',lwd=2)
abline(lm(Adjusted.rel~Adjusted.nifty,data=rt)
        ,col='blue',lwd=2,lty=2)
```

nifty	'zoo' series fr…
rel	'zoo' series fr…
risk_…	0.0001643835616…
rt	'zoo' series fr…

```
+       ,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)
>
```

So, calculating risk premium and then drawing the plots of risk premium grids are being drawn. And then here are some ab lines and then. So, the fitted line this is the fitted line using OLS method.

(Refer Slide Time: 02:44)



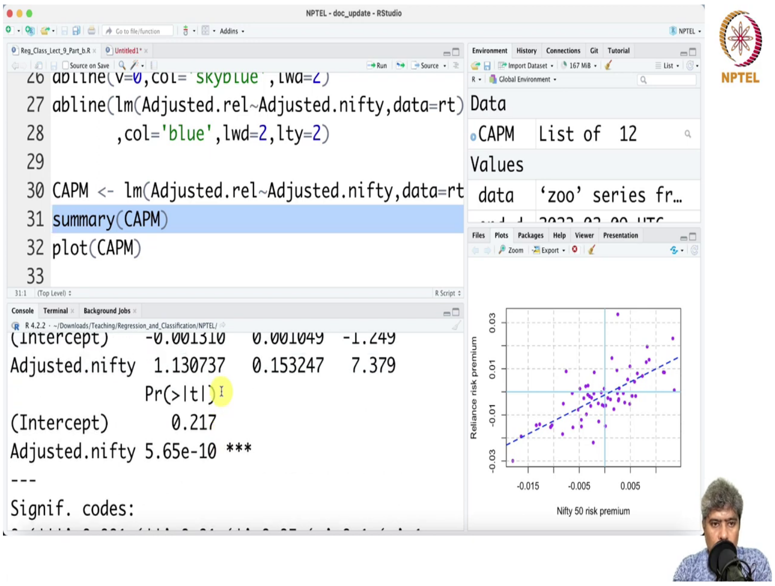
```r
abline(v=0,col='skyblue',lwd=2)
abline(lm(Adjusted.rel~Adjusted.nifty,data=rt)
        ,col='blue',lwd=2,lty=2)

CAPM <- lm(Adjusted.rel~Adjusted.nifty,data=rt
summary(CAPM)
plot(CAPM)
```

```
Data
CAPM    List of 12
Values
data    'zoo' series fr…
```

```
> 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)
> CAPM <- lm(Adjusted.rel~Adjusted.nifty,data=rt)
> summary(CAPM)
```

So, here I am fitting CAPM using OLS method and here is the summary statistics, so ok.

(Refer Slide Time: 02:52)



```
abline(v=0,col='skyblue',lwd=2)
abline(lm(Adjusted.rel~Adjusted.nifty,data=rt)
        ,col='blue',lwd=2,lty=2)

CAPM <- lm(Adjusted.rel~Adjusted.nifty,data=rt
summary(CAPM)
plot(CAPM)
```

Data

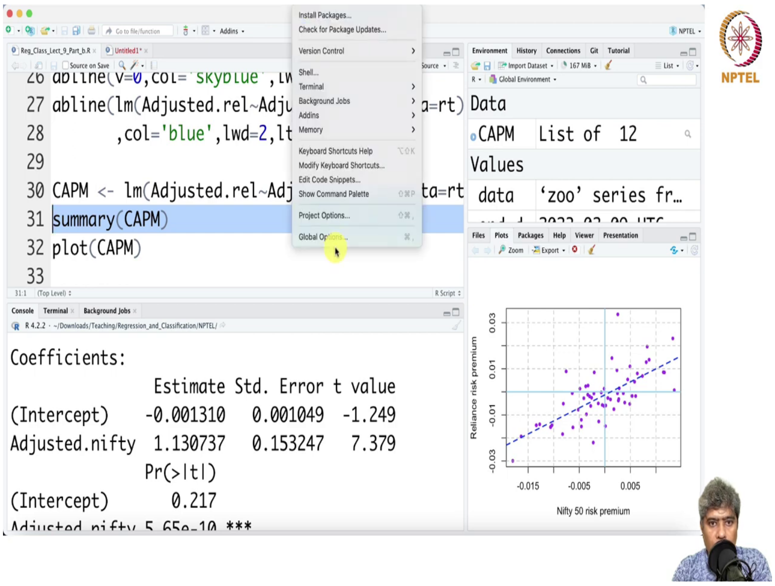
CAPM	List of 12

Values

data	'zoo' series fr…

```
(Intercept)     -0.001310   0.001049  -1.249
Adjusted.nifty  1.130737   0.153247   7.379
                Pr(>|t|)
(Intercept)        0.217
Adjusted.nifty 5.65e-10 ***
---
Signif. codes:
```

(Refer Slide Time: 02:57)



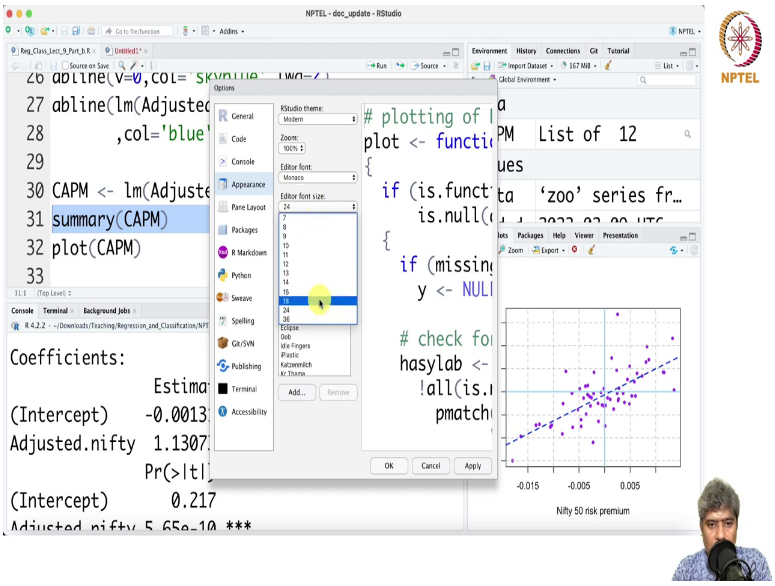
```
abline(v=0,col='skyblue',lw
abline(lm(Adjusted.rel~Adju                a=rt)
        ,col='blue',lwd=2,lt

CAPM <- lm(Adjusted.rel~Adj                ta=rt
summary(CAPM)
plot(CAPM)
```

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

```
Coefficients:
                  Estimate Std. Error t value
(Intercept)    -0.001310   0.001049  -1.249
Adjusted.nifty  1.130737   0.153247   7.379
                Pr(>|t|)
(Intercept)        0.217
Adjusted.nifty 5.65e-10 ***
```

Data	
CAPM	List of 12
Values	
data	'zoo' series fr…

(Refer Slide Time: 03:01)



NPTEL - doc_update - RStudio

```
abline(v=0,col='skyblue',lwd=2)
abline(lm(Adjusted
        ,col='blue'

CAPM <- lm(Adjuste
summary(CAPM)
plot(CAPM)
```

```
Coefficients:
               Estima
(Intercept)    -0.0013
Adjusted.nifty  1.1307
               Pr(>|t|)
(Intercept)       0.217
Adjusted.nifty 5.65e-10 ***
```

Options

RStudio theme: Modern

Zoom: 100%

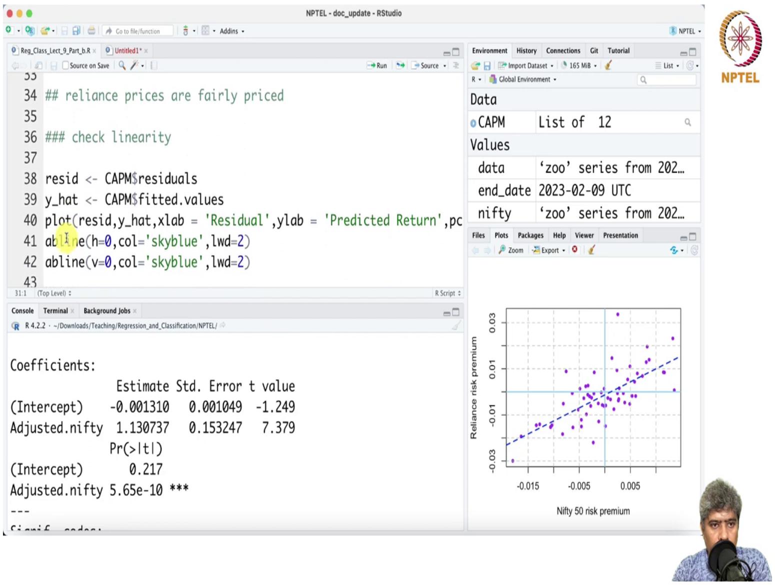
Editor font: Monaco

Editor font size: 24

```
# plotting of l
plot <- functi
{
  if (is.funct
      is.null(
  {
    if (missin
      y <- NULL

    # check fo
    hasylab <-
      !all(is.
        pmatch
```

So, one problem is if we make the font size too big then it will not come as nicely as we would like to see it. So, now it like comes a bit nicely.

(Refer Slide Time: 03:10)

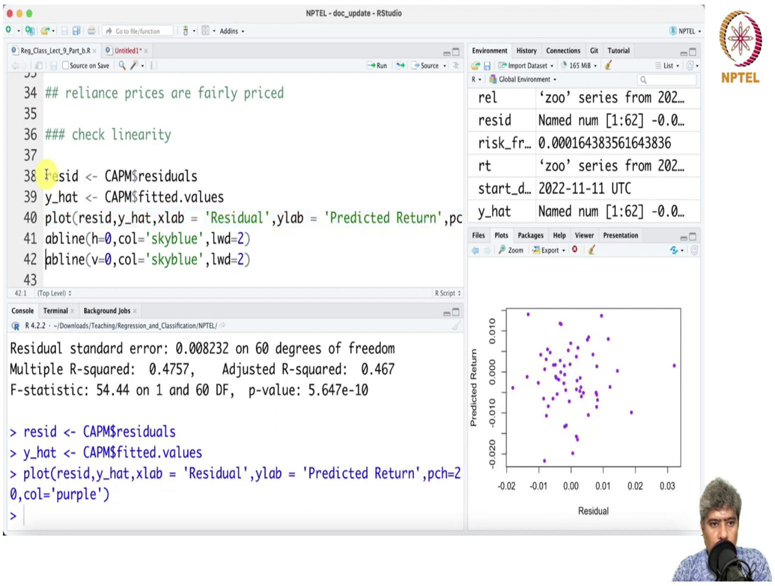


```
## reliance prices are fairly priced

### check linearity

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

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

Coefficients:
                Estimate Std. Error t value
(Intercept)    -0.001310   0.001049  -1.249
Adjusted.nifty  1.130737   0.153247   7.379
               Pr(>|t|)
(Intercept)       0.217
Adjusted.nifty 5.65e-10 ***
---
Signif. codes:
```

Data	
CAPM	List of 12
Values	
data	'zoo' series from 202…
end_date	2023-02-09 UTC
nifty	'zoo' series from 202…

(Refer Slide Time: 03:17)



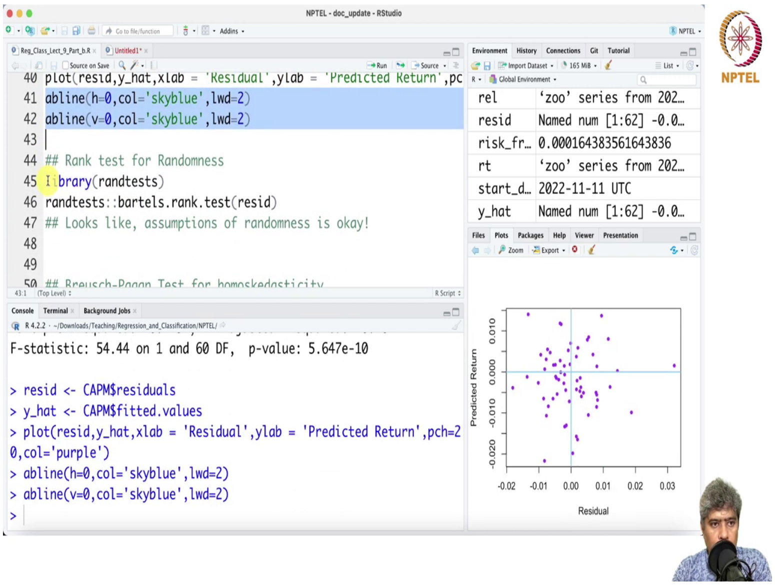
```r
## reliance prices are fairly priced

### check linearity

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

```
Residual standard error: 0.008232 on 60 degrees of freedom
Multiple R-squared:  0.4757,	Adjusted R-squared:  0.467 
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

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

rel	‘zoo’ series from 202…
resid	Named num [1:62] -0.0…
risk_fr…	0.000164383561643836
rt	‘zoo’ series from 202…
start_d…	2022-11-11 UTC
y_hat	Named num [1:62] -0.0…

And then we can take the residuals and then plot the residuals and then can do the Bartels rank test and then we did Breusch Pagan test, rank test and Breusch Pagan test everything was quite good.

(Refer Slide Time: 03:20)

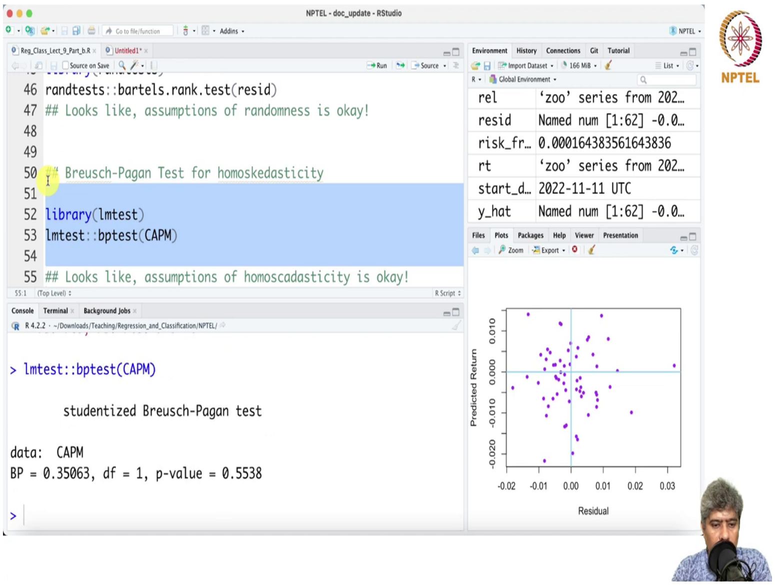


```r
plot(resid,y_hat,xlab = 'Residual',ylab = 'Predicted Return',pc
abline(h=0,col='skyblue',lwd=2)
abline(v=0,col='skyblue',lwd=2)

## Rank test for Randomness
library(randtests)
randtests::bartels.rank.test(resid)
## Looks like, assumptions of randomness is okay!


## Breusch-Pagan Test for homoskedasticity
```

rel	‘zoo’ series from 202…
resid	Named num [1:62] -0.0…
risk_fr…	0.000164383561643836
rt	‘zoo’ series from 202…
start_d…	2022-11-11 UTC
y_hat	Named num [1:62] -0.0…

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

> 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)
> 
```

(Refer Slide Time: 03:28)

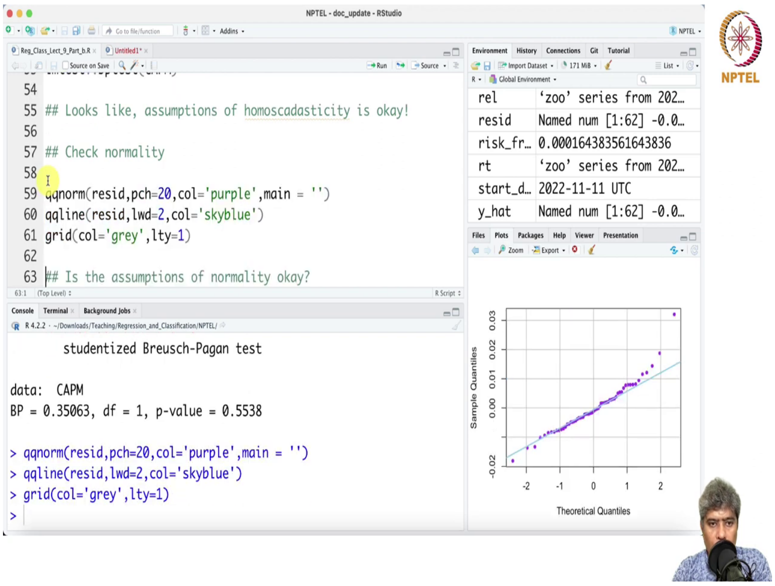


```r
randtests::bartels.rank.test(resid)
## Looks like, assumptions of randomness is okay!


## Breusch-Pagan Test for homoskedasticity

library(lmtest)
lmtest::bptest(CAPM)

## Looks like, assumptions of homoscadasticity is okay!
```

rel	‘zoo’ series from 202…
resid	Named num [1:62] -0.0…
risk_fr…	0.000164383561643836
rt	‘zoo’ series from 202…
start_d…	2022-11-11 UTC
y_hat	Named num [1:62] -0.0…

```
> lmtest::bptest(CAPM)

	studentized Breusch-Pagan test

data:  CAPM
BP = 0.35063, df = 1, p-value = 0.5538

>
```

(Refer Slide Time: 03:34)



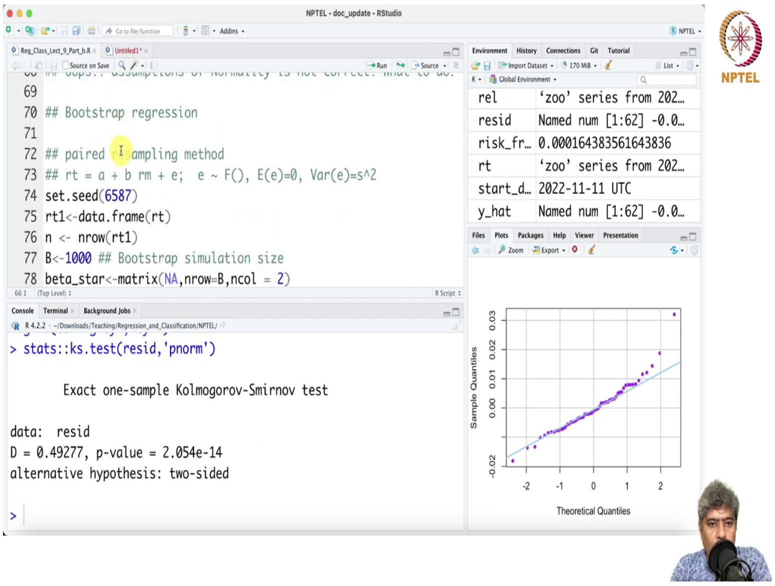
```
## Looks like, assumptions of homoscadasticity is okay!

## Check normality

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

## Is the assumptions of normality okay?
```

```
        studentized Breusch-Pagan test

data:  CAPM
BP = 0.35063, df = 1, p-value = 0.5538

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

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

Then when we did a test for normality it was kind of get rejected and that is where the whole problem starts.

(Refer Slide Time: 03:44)



```
## Oops!: assumptions of Normality is not correct. What to do.

## Bootstrap regression

## paired resampling method
## rt = a + b rm + e;  e ~ F(), E(e)=0, Var(e)=s^2
set.seed(6587)
rt1<-data.frame(rt)
n <- nrow(rt1)
B<-1000 ## Bootstrap simulation size
beta_star<-matrix(NA,nrow=B,ncol = 2)
```

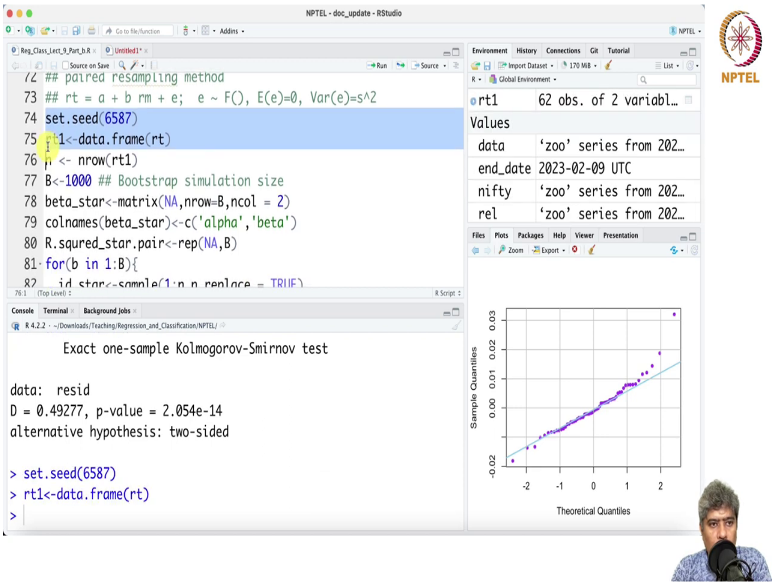
```
> stats::ks.test(resid,'pnorm')

	Exact one-sample Kolmogorov-Smirnov test

data:  resid
D = 0.49277, p-value = 2.054e-14
alternative hypothesis: two-sided

>
```

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

So, we decided to do a bootstrap regression first we are here we are going to apply paired resampling method or paired bootstrap regression method. So, here we are assuming risk premium equal to a alpha plus beta a plus b times market premium plus error. And we are assuming this idiosyncratic return or E is following some distribution. We do not know what is that distribution?

Some distribution F with expected value of E to be 0 and variance of E to be sigma square, but we do not know what is that distribution is this could be any distribution.

(Refer Slide Time: 04:27)

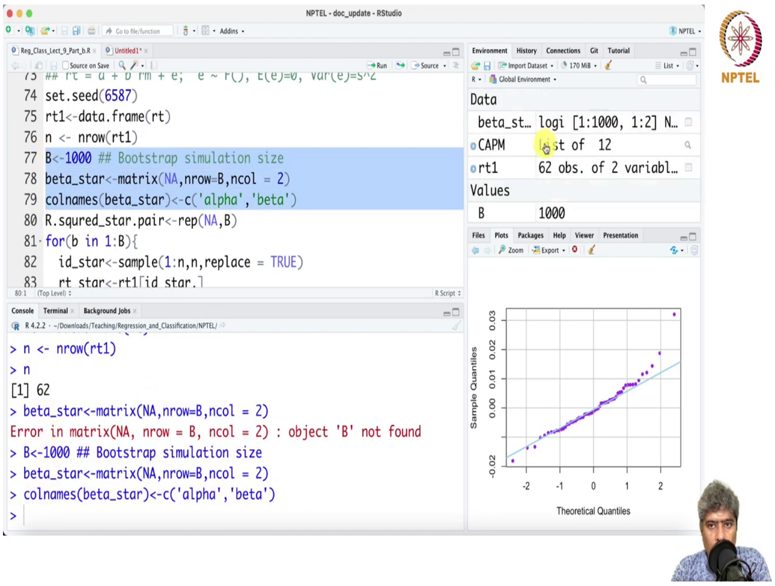


```r
## paired resampling method
## rt = a + b rm + e;  e ~ F(), E(e)=0, Var(e)=s^2
set.seed(6587)
rt1<-data.frame(rt)
n <- nrow(rt1)
B<-1000 ## Bootstrap simulation size
beta_star<-matrix(NA,nrow=B,ncol = 2)
colnames(beta_star)<-c('alpha','beta')
R.squred_star.pair<-rep(NA,B)
for(b in 1:B){
  id_star<-sample(1:n,n,replace = TRUE)
```

```
rt1        62 obs. of 2 variabl…
Values
 data      ‘zoo’ series from 202…
 end_date  2023-02-09 UTC
 nifty     ‘zoo’ series from 202…
 rel       ‘zoo’ series from 202…
```

```
        Exact one-sample Kolmogorov-Smirnov test

data:  resid
D = 0.49277, p-value = 2.054e-14
alternative hypothesis: two-sided

> set.seed(6587)
> rt1<-data.frame(rt)
>
```

(Refer Slide Time: 04:34)



NPTEL - doc_update - RStudio

Reg_Class_Lect_9_Part_b.R   Untitled1*

```r
## rt = a + b rm + e;  e ~ F(), E(e)=0, Var(e)=s^2
set.seed(6587)
rt1<-data.frame(rt)
n <- nrow(rt1)
B<-1000 ## Bootstrap simulation size
beta_star<-matrix(NA,nrow=B,ncol = 2)
colnames(beta_star)<-c('alpha','beta')
R.squred_star.pair<-rep(NA,B)
for(b in 1:B){
  id_star<-sample(1:n,n,replace = TRUE)
  rt_star<-rt1[id_star,]
```

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

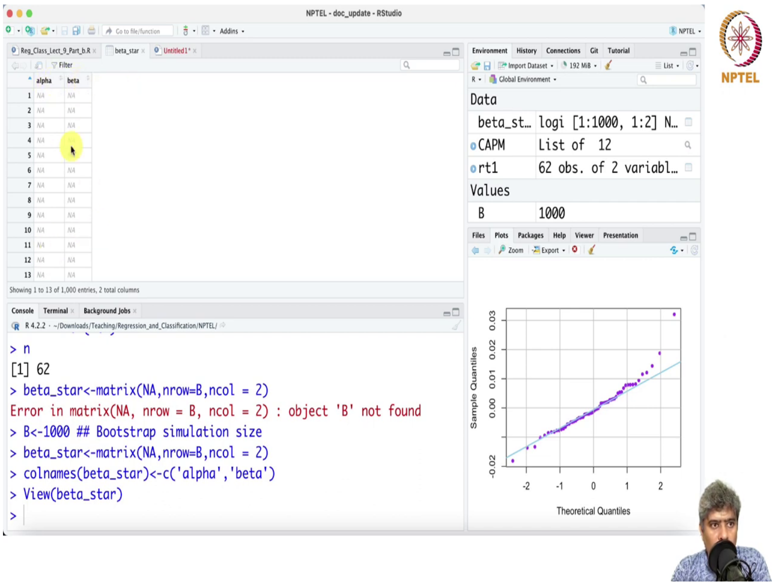
```
> n <- nrow(rt1)
> n
[1] 62
> beta_star<-matrix(NA,nrow=B,ncol = 2)
Error in matrix(NA, nrow = B, ncol = 2) : object 'B' not found
> B<-1000 ## Bootstrap simulation size
> beta_star<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star)<-c('alpha','beta')
>
```

Global Environment

```
Data
  beta_st…   logi [1:1000, 1:2] N…
  CAPM       List of  12
  rt1        62 obs. of 2 variabl…
Values
  B          1000
```

So, we put it as a data frame here is the number of samples is 62, we have taken bootstrap simulation size to be 1000, you can increase it to any number 10,000 or anything. And then I am defining a beta star a matrix of sample simulation size n and number of columns is beta 2, because I am going to estimate alpha and beta. So, oh sorry I have to run this ok.

(Refer Slide Time: 05:03)



	alpha	beta
1	NA	NA
2	NA	NA
3	NA	NA
4	NA	NA
5	NA	NA
6	NA	NA
7	NA	NA
8	NA	NA
9	NA	NA
10	NA	NA
11	NA	NA
12	NA	NA
13	NA	NA

Showing 1 to 13 of 1,000 entries, 2 total columns

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
> n
[1] 62
> beta_star<-matrix(NA,nrow=B,ncol = 2)
Error in matrix(NA, nrow = B, ncol = 2) : object 'B' not found
> B<-1000 ## Bootstrap simulation size
> beta_star<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star)<-c('alpha','beta')
> View(beta_star)
>
```

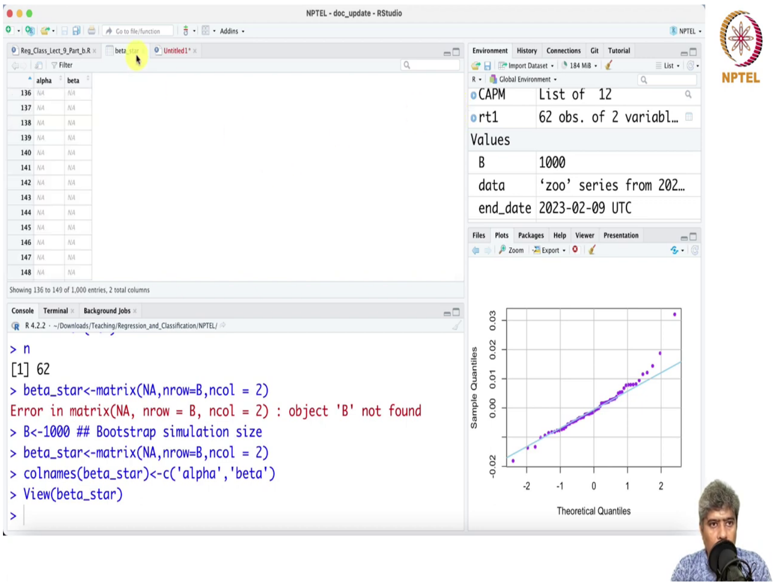
Data

beta_st…	logi [1:1000, 1:2] N…
CAPM	List of 12
rt1	62 obs. of 2 variabl…

Values

B	1000

(Refer Slide Time: 05:15)



	alpha	beta
136	NA	NA
137	NA	NA
138	NA	NA
139	NA	NA
140	NA	NA
141	NA	NA
142	NA	NA
143	NA	NA
144	NA	NA
145	NA	NA
146	NA	NA
147	NA	NA
148	NA	NA

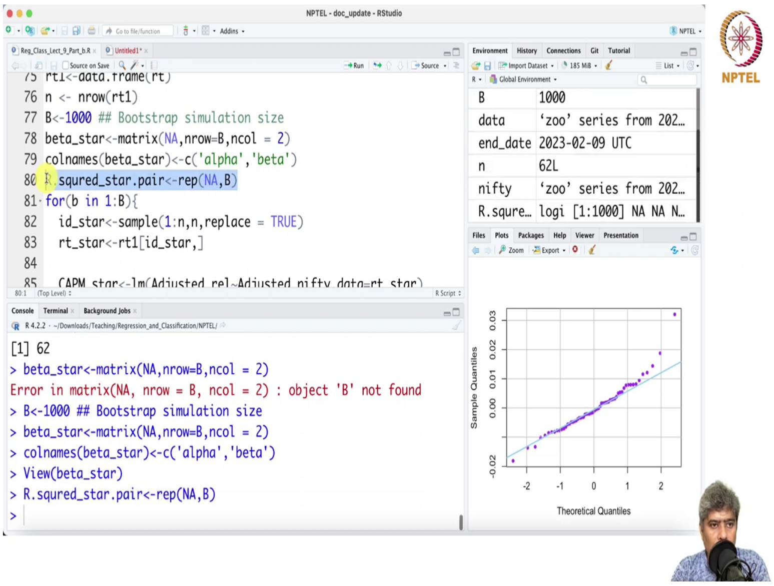
Showing 136 to 149 of 1,000 entries, 2 total columns

CAPM	List of 12
rt1	62 obs. of 2 variabl…
Values	
B	1000
data	‘zoo’ series from 202…
end_date	2023-02-09 UTC

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
> n
[1] 62
> beta_star<-matrix(NA,nrow=B,ncol = 2)
Error in matrix(NA, nrow = B, ncol = 2) : object 'B' not found
> B<-1000 ## Bootstrap simulation size
> beta_star<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star)<-c('alpha','beta')
> View(beta_star)
>
```

Now, if you see I have created this matrix with column name alpha beta and it has size of it has basically if you just go up 10th about 1000 rows number of simulation.

(Refer Slide Time: 05:22)

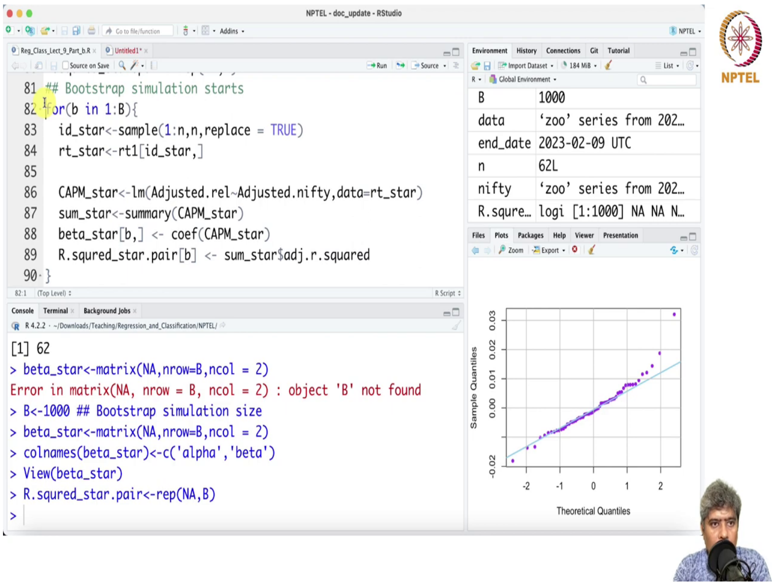


```r
rt1<-data.frame(rt)
n <- nrow(rt1)
B<-1000 ## Bootstrap simulation size
beta_star<-matrix(NA,nrow=B,ncol = 2)
colnames(beta_star)<-c('alpha','beta')
R.squred_star.pair<-rep(NA,B)
for(b in 1:B){
  id_star<-sample(1:n,n,replace = TRUE)
  rt_star<-rt1[id_star,]

  CAPM_star<-lm(Adjusted_rel~Adjusted_nifty,data=rt_star)
```

```
[1] 62
> beta_star<-matrix(NA,nrow=B,ncol = 2)
Error in matrix(NA, nrow = B, ncol = 2) : object 'B' not found
> B<-1000 ## Bootstrap simulation size
> beta_star<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star)<-c('alpha','beta')
> View(beta_star)
> R.squred_star.pair<-rep(NA,B)
>
```

B	1000
data	'zoo' series from 202…
end_date	2023-02-09 UTC
n	62L
nifty	'zoo' series from 202…
R.squre…	logi [1:1000] NA NA N…

(Refer Slide Time: 05:33)



```
## Bootstrap simulation starts
for(b in 1:B){
  id_star<-sample(1:n,n,replace = TRUE)
  rt_star<-rt1[id_star,]

  CAPM_star<-lm(Adjusted.rel~Adjusted.nifty,data=rt_star)
  sum_star<-summary(CAPM_star)
  beta_star[b,] <- coef(CAPM_star)
  R.squred_star.pair[b] <- sum_star$adj.r.squared
}
```

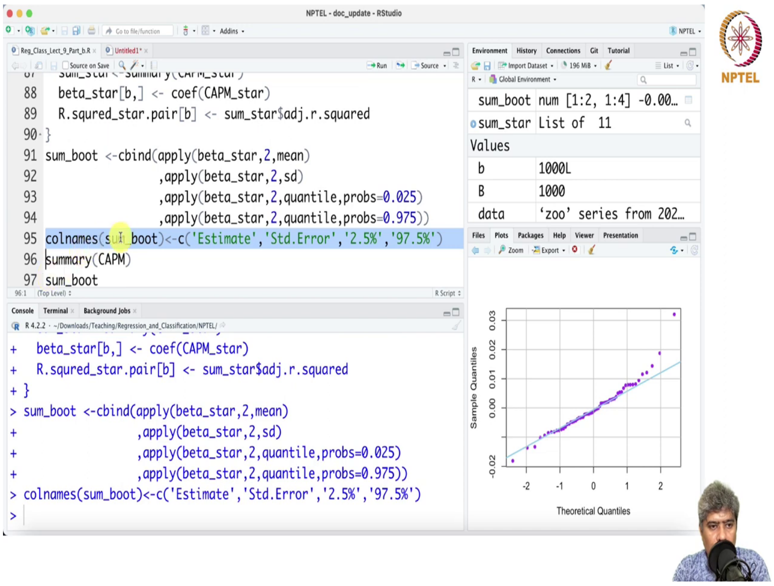
```
[1] 62
> beta_star<-matrix(NA,nrow=B,ncol = 2)
Error in matrix(NA, nrow = B, ncol = 2) : object 'B' not found
> B<-1000 ## Bootstrap simulation size
> beta_star<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star)<-c('alpha','beta')
> View(beta_star)
> R.squred_star.pair<-rep(NA,B)
>
```

B	1000
data	‘zoo’ series from 202…
end_date	2023-02-09 UTC
n	62L
nifty	‘zoo’ series from 202…
R.squre…	logi [1:1000] NA NA N…

Now, I am going to also create another place for R square I am calling it R square star. Then here my loop start bootstrap regression bootstrap simulation starts bootstrap simulation starts. So, first from 1 is to n id I am going to draw ransom’s id I am going to draw the random ids and those ids I am going to take the samples random samples from the samples itself.

So, rt 1 is the sample itself from there I am going to draw random samples and those are that is the rt star and from the rt star I am going to you know fit the model and from the model I am going to calculate the summary and from the summary I am going to I am extracting the coefficients alpha beta ok and the adjusted R squared I am keeping it here.

(Refer Slide Time: 06:40)



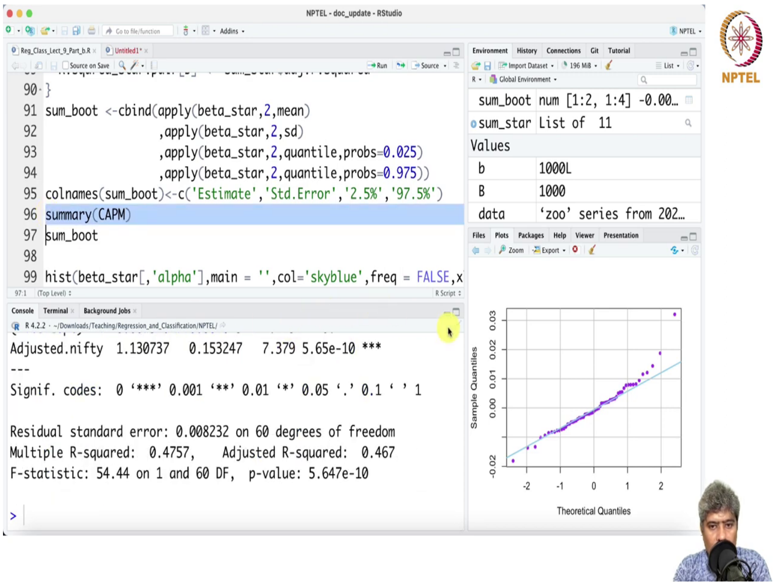
```r
  sum_star <-summary(CAPM_star)
  beta_star[b,] <- coef(CAPM_star)
  R.squred_star.pair[b] <- sum_star$adj.r.squared
}
sum_boot <-cbind(apply(beta_star,2,mean)
                 ,apply(beta_star,2,sd)
                 ,apply(beta_star,2,quantile,probs=0.025)
                 ,apply(beta_star,2,quantile,probs=0.975))
colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
summary(CAPM)
sum_boot
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
+   beta_star[b,] <- coef(CAPM_star)
+   R.squred_star.pair[b] <- sum_star$adj.r.squared
+ }
> sum_boot <-cbind(apply(beta_star,2,mean)
+                  ,apply(beta_star,2,sd)
+                  ,apply(beta_star,2,quantile,probs=0.025)
+                  ,apply(beta_star,2,quantile,probs=0.975))
> colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
>
```

```
sum_boot  num [1:2, 1:4] -0.00…
sum_star  List of  11
Values
b         1000L
B         1000
data      'zoo' series from 202…
```

So, let me just run this piece of code and it is run in almost no time and then I am going to. So, summarize the bootstrap statistics here so few and then estimate standard error and 25 percent.

(Refer Slide Time: 06:57)



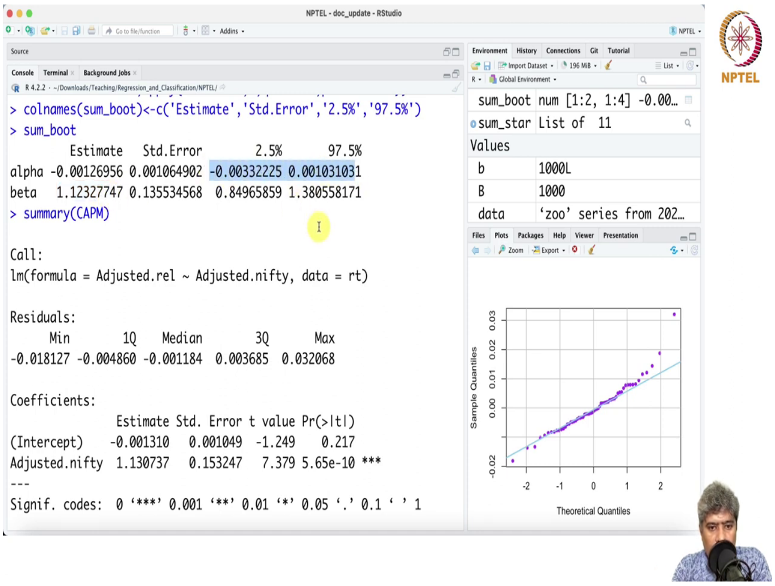
```r
}
sum_boot <-cbind(apply(beta_star,2,mean)
                 ,apply(beta_star,2,sd)
                 ,apply(beta_star,2,quantile,probs=0.025)
                 ,apply(beta_star,2,quantile,probs=0.975))
colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
summary(CAPM)
sum_boot

hist(beta_star[,'alpha'],main = '',col='skyblue',freq = FALSE,x
```

```
Adjusted.nifty  1.130737   0.153247   7.379 5.65e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.008232 on 60 degrees of freedom
Multiple R-squared:  0.4757,    Adjusted R-squared:  0.467
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

>
```

sum_boot	num [1:2, 1:4] -0.00…
sum_star	List of 11
Values	
b	1000L
B	1000
data	‘zoo’ series from 202…

So, now if you run this, so estimates are like negative 0.0012 and 0.12. So, now you can see the alpha is containing the 0 the confidence interval this is the confidence interval for alpha and it contain 0. And similarly, this is the confidence interval for beta it contains 1. So, for this is for reliance and remember that summary statistics for CAPM this is the OLS thing.

(Refer Slide Time: 07:28)

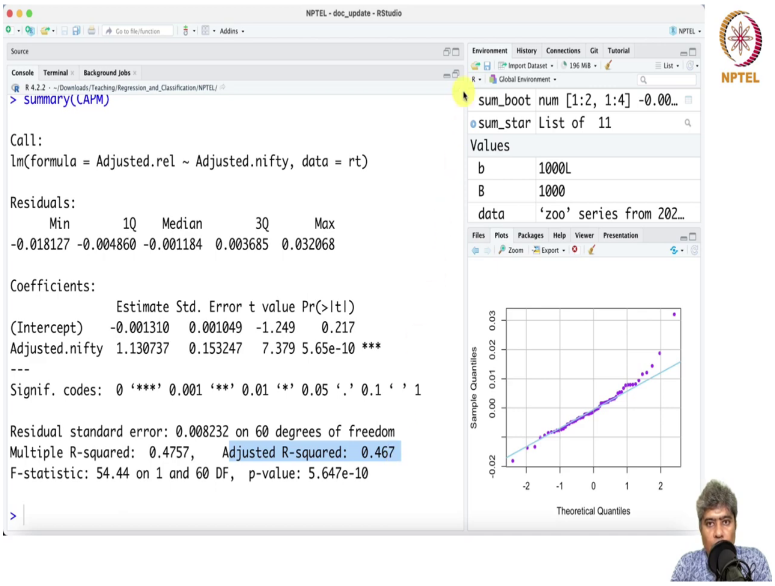


```
> colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
> sum_boot
         Estimate   Std.Error        2.5%       97.5%
alpha -0.00126956 0.001064902 -0.00332225 0.001031031
beta   1.12327747 0.135534568  0.84965859 1.380558171
> summary(CAPM)

Call:
lm(formula = Adjusted.rel ~ Adjusted.nifty, data = rt)

Residuals:
      Min        1Q    Median        3Q       Max
-0.018127 -0.004860 -0.001184  0.003685  0.032068

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.001310   0.001049  -1.249    0.217
Adjusted.nifty  1.130737   0.153247   7.379 5.65e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
sum_boot  num [1:2, 1:4] -0.00…
sum_star  List of  11
Values
b         1000L
B         1000
data      ‘zoo’ series from 202…
```

So, if you now carefully look into it that OLS estimate is 0.0013 and if you round it of to 4 decimal places it is actually negative 0.0013, beta is 1.1307 1 this is beta here 1.12327 the this is the coefficient.

(Refer Slide Time: 07:56)

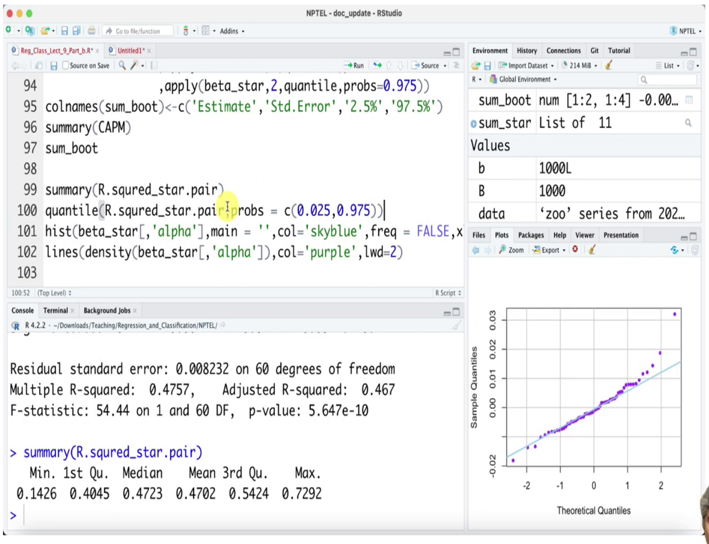


```
> summary(CAPM)

Call:
lm(formula = Adjusted.rel ~ Adjusted.nifty, data = rt)

Residuals:
      Min        1Q    Median        3Q       Max
-0.018127 -0.004860 -0.001184  0.003685  0.032068

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.001310   0.001049  -1.249    0.217
Adjusted.nifty  1.130737   0.153247   7.379 5.65e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.008232 on 60 degrees of freedom
Multiple R-squared:  0.4757,    Adjusted R-squared:  0.467
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

>
```

```
sum_boot  num [1:2, 1:4] -0.00…
sum_star  List of  11
Values
b         1000L
B         1000
data      ‘zoo’ series from 202…
```

So, now since the underlying distribution does not or underlying model assumption does not hold good, so these inference based on t value and p value is not valid inference. But here I am assuming that the underlying distribution is just unknown and I am doing a non-parametric test. So, these inference is valid though there is not much change in my inference, but this inference is valid.

(Refer Slide Time: 08:41)



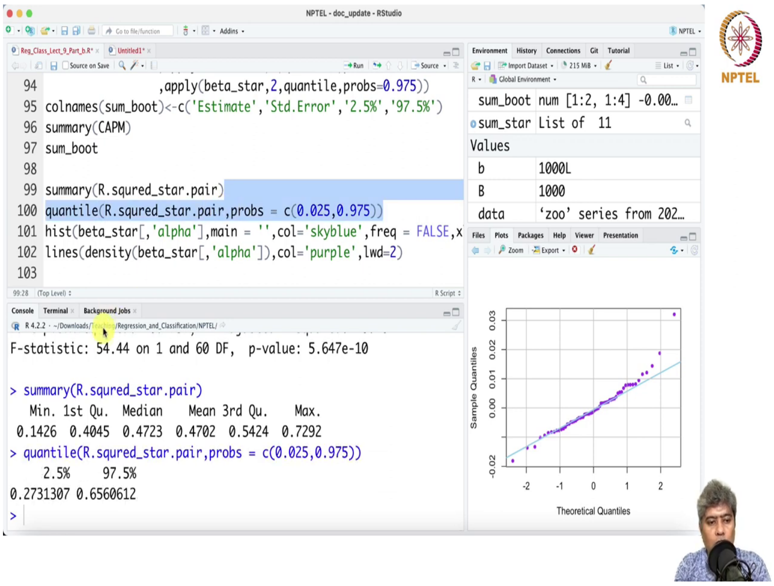
```
                 ,apply(beta_star,2,quantile,probs=0.975))
colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
summary(CAPM)
sum_boot

summary(R.squred_star.pair)
quantile(R.squred_star.pair,probs = c(0.025,0.975))
hist(beta_star[,'alpha'],main = '',col='skyblue',freq = FALSE,x
lines(density(beta_star[,'alpha']),col='purple',lwd=2)
```

```
sum_boot   num [1:2, 1:4] -0.00…
sum_star   List of 11
Values
b          1000L
B          1000
data       'zoo' series from 202…
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

Residual standard error: 0.008232 on 60 degrees of freedom
Multiple R-squared:  0.4757,	Adjusted R-squared:  0.467 
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

> summary(R.squred_star.pair)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.1426  0.4045  0.4723  0.4702  0.5424  0.7292 
> 
```

Here the adjusted R square is given 0.467, but what we can do is essentially we can just say summary of this. So, this is the bootstrap statistics for the R square adjusted R square and we can quantile say we can create a quantile of R squared with probability equal to 0.025 comma 0.975.

(Refer Slide Time: 09:10)



NPTEL - doc_update - RStudio

Reg_Class_Lect_9_Part_b.R* Untitled1*

```
                         ,apply(beta_star,2,quantile,probs=0.975))
colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
summary(CAPM)
sum_boot

summary(R.squred_star.pair)
quantile(R.squred_star.pair,probs = c(0.025,0.975))
hist(beta_star[,'alpha'],main = '',col='skyblue',freq = FALSE,x
lines(density(beta_star[,'alpha']),col='purple',lwd=2)
```

R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

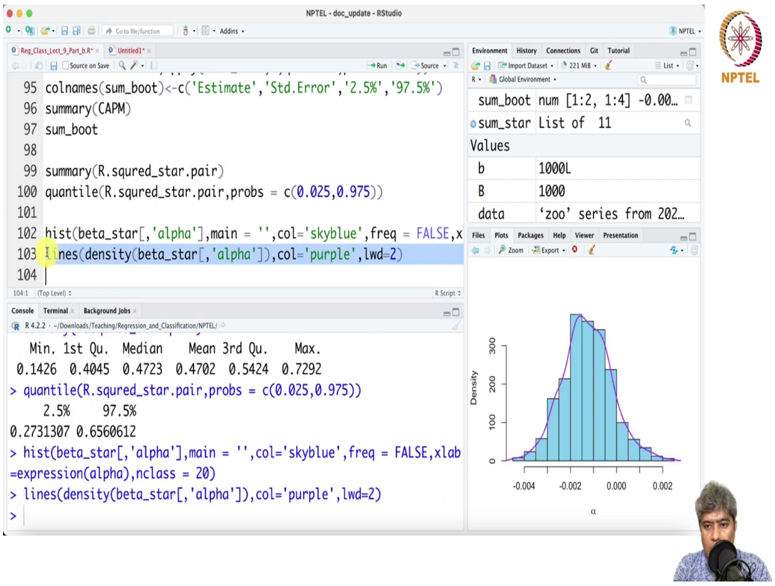
```
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

> summary(R.squred_star.pair)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.1426  0.4045  0.4723  0.4702  0.5424  0.7292 
> quantile(R.squred_star.pair,probs = c(0.025,0.975))
     2.5%     97.5% 
0.2731307 0.6560612 
> 
```

```
sum_boot   num [1:2, 1:4] -0.00…
sum_star   List of  11
Values
b          1000L
B          1000
data       'zoo' series from 202…
```

Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Sample Quantiles
-0.02
0.00
0.01
0.02
0.03
-2
-1
0
1
2
Theoretical Quantiles
NPTEL

Now, this gives me a 95 percent confidence interval for adjusted R square. Now, in typically it is very difficult to get a any statistical inference for R square, but using bootstrap we can calculate a 95 percent confidence interval for adjusted R square.

(Refer Slide Time: 09:34)



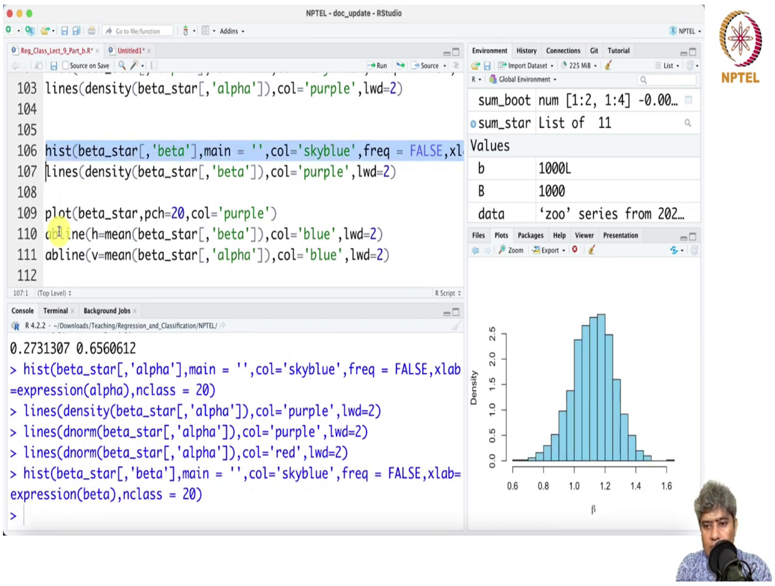
```
colnames(sum_boot)<-c('Estimate','Std.Error','2.5%','97.5%')
summary(CAPM)
sum_boot

summary(R.squred_star.pair)
quantile(R.squred_star.pair,probs = c(0.025,0.975))

hist(beta_star[,'alpha'],main = '',col='skyblue',freq = FALSE,x
lines(density(beta_star[,'alpha']),col='purple',lwd=2)
```

```
sum_boot   num [1:2, 1:4] -0.00…
sum_star   List of  11
Values
b          1000L
B          1000
data       'zoo' series from 202…
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.1426  0.4045  0.4723  0.4702  0.5424  0.7292 
> quantile(R.squred_star.pair,probs = c(0.025,0.975))
     2.5%     97.5% 
0.2731307 0.6560612 
> hist(beta_star[,'alpha'],main = '',col='skyblue',freq = FALSE,xlab
=expression(alpha),nclass = 20)
> lines(density(beta_star[,'alpha']),col='purple',lwd=2)
> 
```

So, this is a very interesting thing. So, you can here we have the alpha the sampling this is bootstrap sampling distribution of alpha and here is the density that we are drawing and then if you would have lines.

(Refer Slide Time: 09:57)

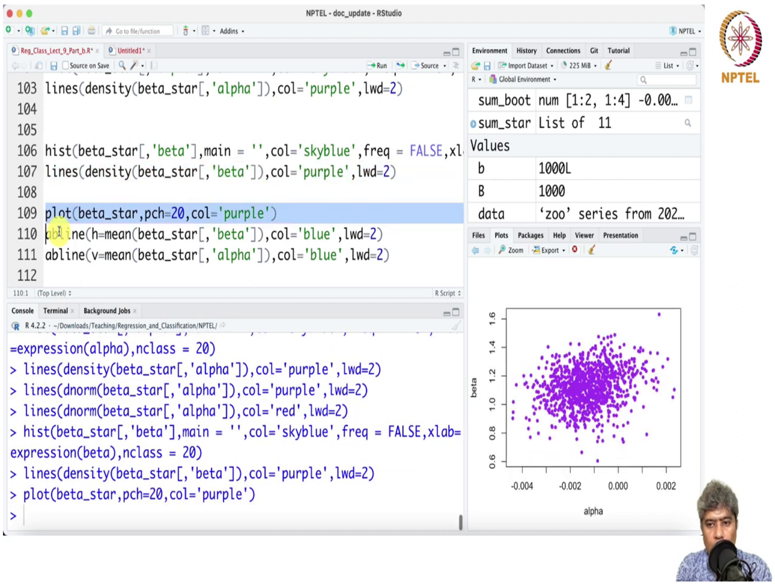


```
lines(density(beta_star[,'alpha']),col='purple',lwd=2)


hist(beta_star[,'beta'],main = '',col='skyblue',freq = FALSE,xl
lines(density(beta_star[,'beta']),col='purple',lwd=2)

plot(beta_star,pch=20,col='purple')
abline(h=mean(beta_star[,'beta']),col='blue',lwd=2)
abline(v=mean(beta_star[,'alpha']),col='blue',lwd=2)
```

```
0.2731307 0.6560612
> hist(beta_star[,'alpha'],main = '',col='skyblue',freq = FALSE,xlab
=expression(alpha),nclass = 20)
> lines(density(beta_star[,'alpha']),col='purple',lwd=2)
> lines(dnorm(beta_star[,'alpha']),col='purple',lwd=2)
> lines(dnorm(beta_star[,'alpha']),col='red',lwd=2)
> hist(beta_star[,'beta'],main = '',col='skyblue',freq = FALSE,xlab=
expression(beta),nclass = 20)
>
```

```
sum_boot  num [1:2, 1:4] -0.00…
sum_star  List of  11
Values
b         1000L
B         1000
data      'zoo' series from 202…
```

(Refer Slide Time: 10:03)



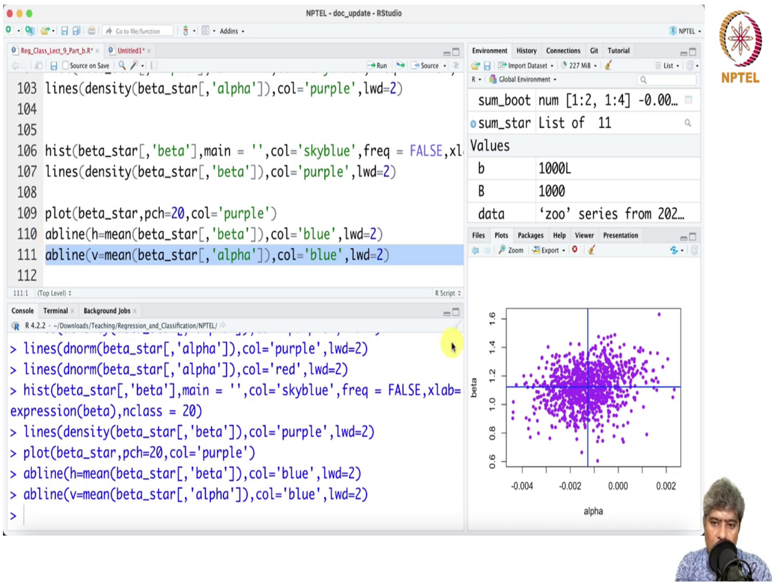
```r
lines(density(beta_star[,'alpha']),col='purple',lwd=2)


hist(beta_star[,'beta'],main = '',col='skyblue',freq = FALSE,xl
lines(density(beta_star[,'beta']),col='purple',lwd=2)

plot(beta_star,pch=20,col='purple')
abline(h=mean(beta_star[,'beta']),col='blue',lwd=2)
abline(v=mean(beta_star[,'alpha']),col='blue',lwd=2)
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
=expression(alpha),nclass = 20)
> lines(density(beta_star[,'alpha']),col='purple',lwd=2)
> lines(dnorm(beta_star[,'alpha']),col='purple',lwd=2)
> lines(dnorm(beta_star[,'alpha']),col='red',lwd=2)
> hist(beta_star[,'beta'],main = '',col='skyblue',freq = FALSE,xlab=
expression(beta),nclass = 20)
> lines(density(beta_star[,'beta']),col='purple',lwd=2)
> plot(beta_star,pch=20,col='purple')
>
```

```
sum_boot   num [1:2, 1:4] -0.00…
sum_star   List of  11
Values
b          1000L
B          1000
data       'zoo' series from 202…
```

And similarly, we can do the histogram of the this is the beta and this is the lines.

(Refer Slide Time: 10:15)



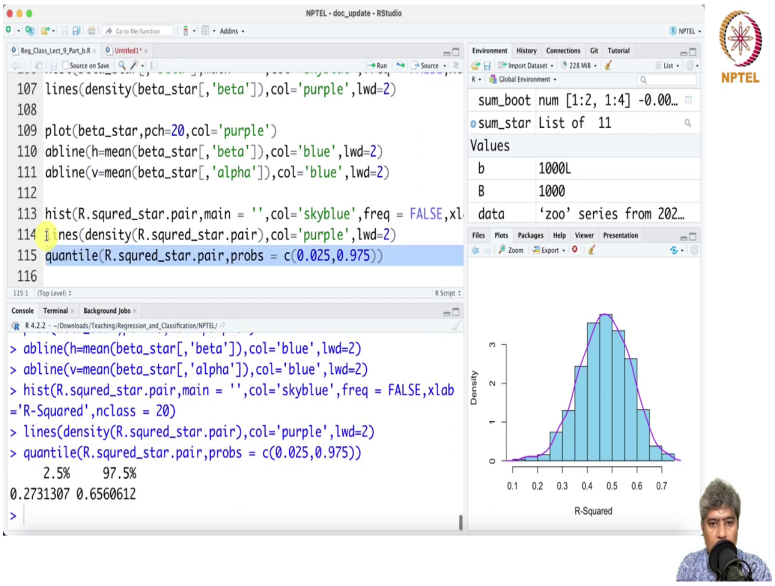
```
lines(density(beta_star[,'alpha']),col='purple',lwd=2)


hist(beta_star[,'beta'],main = '',col='skyblue',freq = FALSE,xl
lines(density(beta_star[,'beta']),col='purple',lwd=2)

plot(beta_star,pch=20,col='purple')
abline(h=mean(beta_star[,'beta']),col='blue',lwd=2)
abline(v=mean(beta_star[,'alpha']),col='blue',lwd=2)
```

```
sum_boot  num [1:2, 1:4] -0.00…
sum_star  List of  11
Values
b         1000L
B         1000
data      'zoo' series from 202…
```

```
> lines(dnorm(beta_star[,'alpha']),col='purple',lwd=2)
> lines(dnorm(beta_star[,'alpha']),col='red',lwd=2)
> hist(beta_star[,'beta'],main = '',col='skyblue',freq = FALSE,xlab=
expression(beta),nclass = 20)
> lines(density(beta_star[,'beta']),col='purple',lwd=2)
> plot(beta_star,pch=20,col='purple')
> abline(h=mean(beta_star[,'beta']),col='blue',lwd=2)
> abline(v=mean(beta_star[,'alpha']),col='blue',lwd=2)
>
```

If we do the beta star this is the distribution between joint distribution of sampling distribution of alpha and beta. And we see that there is not much correlation that we are seeing between alpha and beta which is a good actually.

(Refer Slide Time: 10:26)



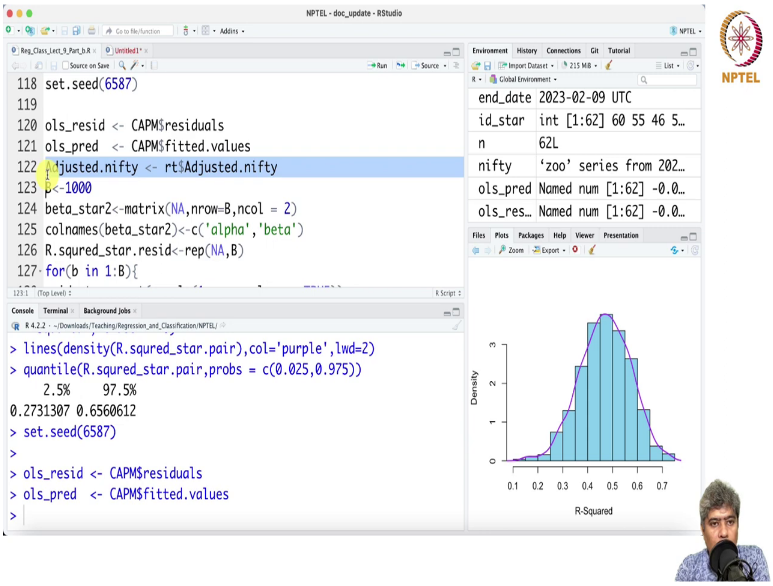
```
lines(density(beta_star[,'beta']),col='purple',lwd=2)

plot(beta_star,pch=20,col='purple')
abline(h=mean(beta_star[,'beta']),col='blue',lwd=2)
abline(v=mean(beta_star[,'alpha']),col='blue',lwd=2)

hist(R.squred_star.pair,main = '',col='skyblue',freq = FALSE,xl
lines(density(R.squred_star.pair),col='purple',lwd=2)
quantile(R.squred_star.pair,probs = c(0.025,0.975))
```

```
sum_boot  num [1:2, 1:4] -0.00…
sum_star  List of  11
Values
b         1000L
B         1000
data      'zoo' series from 202…
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
> abline(h=mean(beta_star[,'beta']),col='blue',lwd=2)
> abline(v=mean(beta_star[,'alpha']),col='blue',lwd=2)
> hist(R.squred_star.pair,main = '',col='skyblue',freq = FALSE,xlab
='R-Squared',nclass = 20)
> lines(density(R.squred_star.pair),col='purple',lwd=2)
> quantile(R.squred_star.pair,probs = c(0.025,0.975))
     2.5%     97.5% 
0.2731307 0.6560612 
> 
```

So, here is the sampling distribution of bootstrap sampling distribution of R square here is the lines density; kernel density here is the R square.

(Refer Slide Time: 10:41)



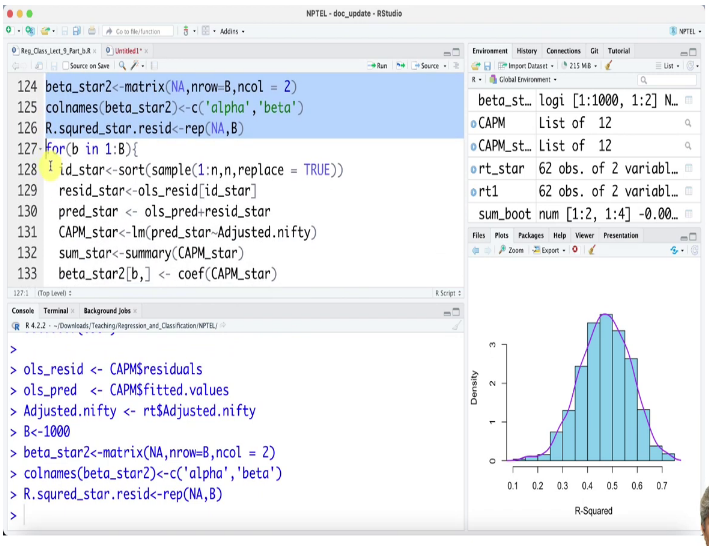
```
set.seed(6587)

ols_resid <- CAPM$residuals
ols_pred  <- CAPM$fitted.values
Adjusted.nifty <- rt$Adjusted.nifty
B<-1000
beta_star2<-matrix(NA,nrow=B,ncol = 2)
colnames(beta_star2)<-c('alpha','beta')
R.squred_star.resid<-rep(NA,B)
for(b in 1:B){
```

```
> lines(density(R.squred_star.pair),col='purple',lwd=2)
> quantile(R.squred_star.pair,probs = c(0.025,0.975))
     2.5%     97.5%
0.2731307 0.6560612
> set.seed(6587)
>
> ols_resid <- CAPM$residuals
> ols_pred  <- CAPM$fitted.values
>
```

end_date	2023-02-09 UTC
id_star	int [1:62] 60 55 46 5…
n	62L
nifty	‘zoo’ series from 202…
ols_pred	Named num [1:62] -0.0…
ols_res…	Named num [1:62] -0.0…

Now, we are going to do the residual resampling or Residual Bootstrap Pregression ok and from the CAPM we just take the OLS residual and OLS predicted values and then this is the adjusted nifty and same way we define the matrixes.

(Refer Slide Time: 11:11)



NPTEL - doc_update - RStudio

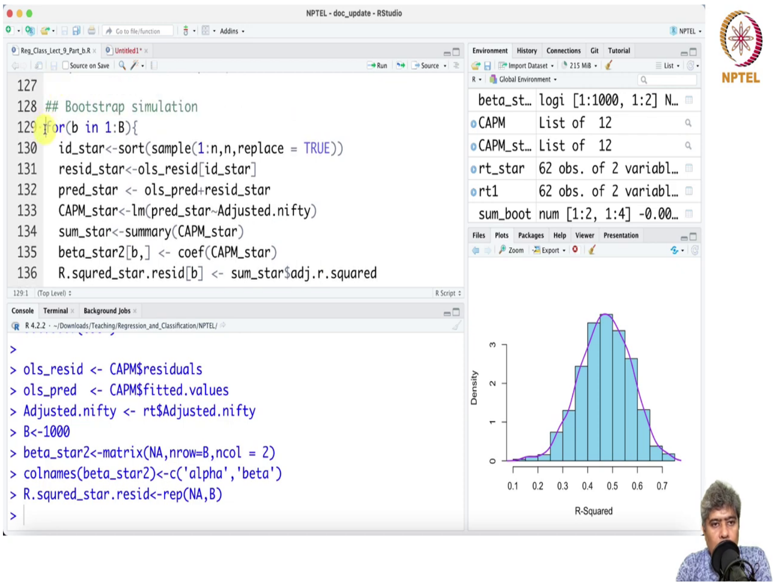
```r
beta_star2<-matrix(NA,nrow=B,ncol = 2)
colnames(beta_star2)<-c('alpha','beta')
R.squred_star.resid<-rep(NA,B)
for(b in 1:B){
  id_star<-sort(sample(1:n,n,replace = TRUE))
  resid_star<-ols_resid[id_star]
  pred_star <- ols_pred+resid_star
  CAPM_star<-lm(pred_star~Adjusted.nifty)
  sum_star<-summary(CAPM_star)
  beta_star2[b,] <- coef(CAPM_star)
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
>
> ols_resid <- CAPM$residuals
> ols_pred  <- CAPM$fitted.values
> Adjusted.nifty <- rt$Adjusted.nifty
> B<-1000
> beta_star2<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star2)<-c('alpha','beta')
> R.squred_star.resid<-rep(NA,B)
>
```

```
beta_st…   logi [1:1000, 1:2] N…
CAPM       List of  12
CAPM_st…   List of  12
rt_star    62 obs. of 2 variabl…
rt1        62 obs. of 2 variabl…
sum_boot   num [1:2, 1:4] -0.00…
```

Now, here we have starting the bootstrap statistic Bootstrap simulation.

(Refer Slide Time: 11:18)



NPTEL - doc_update - RStudio

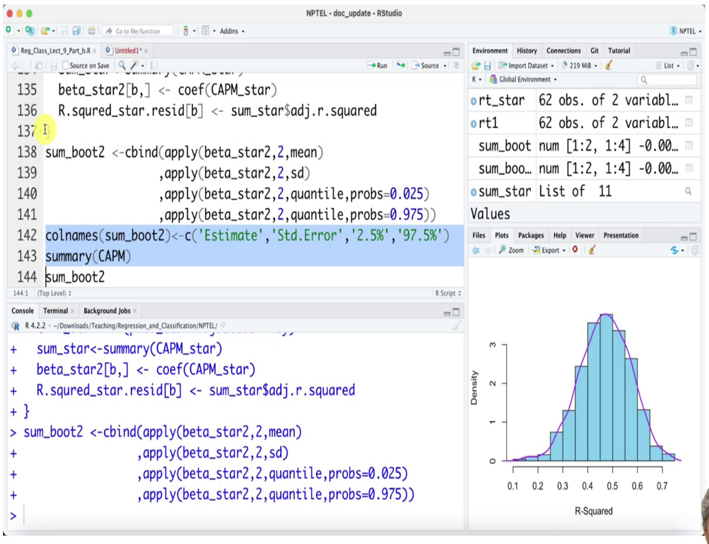
```r
## Bootstrap simulation
for(b in 1:B){
  id_star<-sort(sample(1:n,n,replace = TRUE))
  resid_star<-ols_resid[id_star]
  pred_star <- ols_pred+resid_star
  CAPM_star<-lm(pred_star~Adjusted.nifty)
  sum_star<-summary(CAPM_star)
  beta_star2[b,] <- coef(CAPM_star)
  R.squred_star.resid[b] <- sum_star$adj.r.squared
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
>
> ols_resid <- CAPM$residuals
> ols_pred  <- CAPM$fitted.values
> Adjusted.nifty <- rt$Adjusted.nifty
> B<-1000
> beta_star2<-matrix(NA,nrow=B,ncol = 2)
> colnames(beta_star2)<-c('alpha','beta')
> R.squred_star.resid<-rep(NA,B)
>
```

beta_st…	logi [1:1000, 1:2] N…
CAPM	List of 12
CAPM_st…	List of 12
rt_star	62 obs. of 2 variabl…
rt1	62 obs. of 2 variabl…
sum_boot	num [1:2, 1:4] -0.00…

Now, here is the id star I am from the OLS residual I am just simulating the residuals, then from the predicted ols predicted value you just take the predicted star predicted value, because X beta hat is nothing but the ols predicted value right and then you just add the error and that will give you the Y star. And then based on the Y star you compute the again fitted value and CAPM star and then all you do from there you compute the coefficients and from the and adjusted R square.

(Refer Slide Time: 12:20)



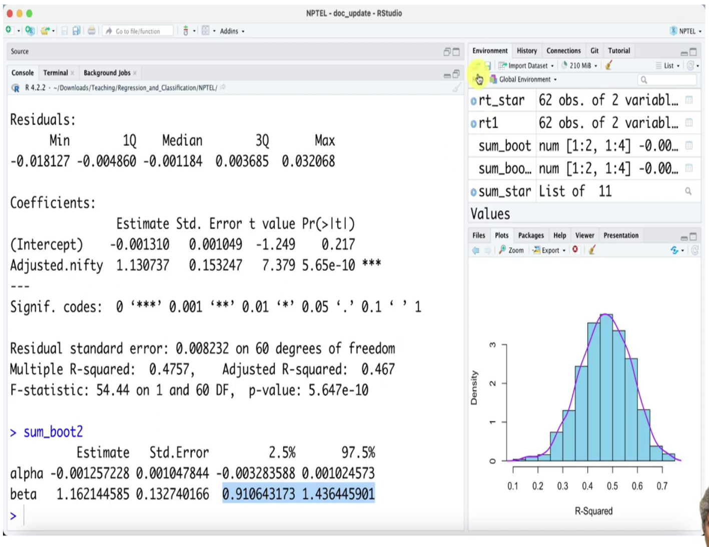
```
  beta_star2[b,] <- coef(CAPM_star)
  R.squred_star.resid[b] <- sum_star$adj.r.squared
}

sum_boot2 <-cbind(apply(beta_star2,2,mean)
                  ,apply(beta_star2,2,sd)
                  ,apply(beta_star2,2,quantile,probs=0.025)
                  ,apply(beta_star2,2,quantile,probs=0.975))
colnames(sum_boot2)<-c('Estimate','Std.Error','2.5%','97.5%')
summary(CAPM)
sum_boot2
```

```
rt_star    62 obs. of 2 variabl…
rt1        62 obs. of 2 variabl…
sum_boot   num [1:2, 1:4] -0.00…
sum_boo…   num [1:2, 1:4] -0.00…
sum_star   List of  11
Values
```

```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
+   sum_star<-summary(CAPM_star)
+   beta_star2[b,] <- coef(CAPM_star)
+   R.squred_star.resid[b] <- sum_star$adj.r.squared
+ }
> sum_boot2 <-cbind(apply(beta_star2,2,mean)
+                   ,apply(beta_star2,2,sd)
+                   ,apply(beta_star2,2,quantile,probs=0.025)
+                   ,apply(beta_star2,2,quantile,probs=0.975))
>
```

So, same way you just do that it is very fast for 1000 simulation it does very quickly actually.

(Refer Slide Time: 12:33)

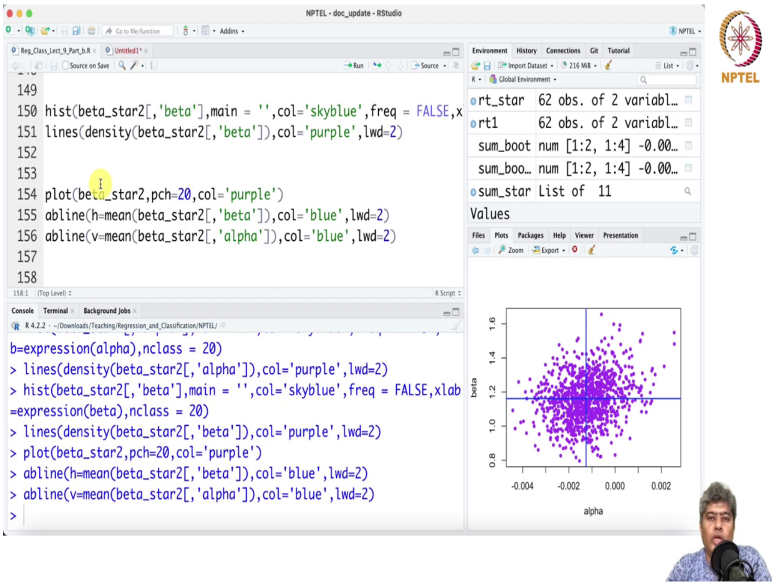


```
Residuals:
      Min        1Q    Median        3Q       Max
-0.018127 -0.004860 -0.001184  0.003685  0.032068

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.001310   0.001049  -1.249    0.217
Adjusted.nifty  1.130737   0.153247   7.379 5.65e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.008232 on 60 degrees of freedom
Multiple R-squared:  0.4757,    Adjusted R-squared:  0.467
F-statistic: 54.44 on 1 and 60 DF,  p-value: 5.647e-10

> sum_boot2
          Estimate   Std.Error         2.5%       97.5%
alpha -0.001257228 0.001047844 -0.003283588 0.001024573
beta   1.162144585 0.132740166  0.910643173 1.436445901
>
```

rt_star	62 obs. of 2 variabl…
rt1	62 obs. of 2 variabl…
sum_boot	num [1:2, 1:4] -0.00…
sum_boo…	num [1:2, 1:4] -0.00…
sum_star	List of 11

So, here is the estimates are pretty much very close to similar previously and confidence interval also does not change much.

(Refer Slide Time: 12:48)



```
hist(beta_star2[,'beta'],main = '',col='skyblue',freq = FALSE,x
lines(density(beta_star2[,'beta']),col='purple',lwd=2)


plot(beta_star2,pch=20,col='purple')
abline(h=mean(beta_star2[,'beta']),col='blue',lwd=2)
abline(v=mean(beta_star2[,'alpha']),col='blue',lwd=2)
```

```
rt_star     62 obs. of 2 variabl…
rt1         62 obs. of 2 variabl…
sum_boot    num [1:2, 1:4] -0.00…
sum_boo…    num [1:2, 1:4] -0.00…
sum_star    List of  11
Values
```

```
b=expression(alpha),nclass = 20)
> lines(density(beta_star2[,'alpha']),col='purple',lwd=2)
> hist(beta_star2[,'beta'],main = '',col='skyblue',freq = FALSE,xlab
=expression(beta),nclass = 20)
> lines(density(beta_star2[,'beta']),col='purple',lwd=2)
> plot(beta_star2,pch=20,col='purple')
> abline(h=mean(beta_star2[,'beta']),col='blue',lwd=2)
> abline(v=mean(beta_star2[,'alpha']),col='blue',lwd=2)
>
```

So, residual bootstrap holds pretty much similar not much change because and this is the alpha distribution of alpha, this is the distribution of beta, this is the joint distribution of alpha and beta when we are doing residual bootstrap regression. So, that is how we do paired bootstrap statistics or bootstrap regression and residual bootstrap regression.

If sometimes what happens as you know that you fit the model and what happens is you fit a model you did a very good job with modeling and do a reasonable predict prediction also. Suppose you have achieved a descent R squared very reasonably low RMSE, but and most likely you are ok to you know accept the model to the production.

Now, people can still re-question that if the model assumption does not hold good. Now, often time what I have seen that if this is the if the case is that randomness is ok and homoscedasticity is ok. Then and but the it is the distribution that is the problem because distribution is always a very strong assumption; then I will always recommend you to resort to bootstrap.

So, if you do the bootstrapping what happens is you still work with the same model only the statistical testing of hypothesis and the inferences. Now, you are doing with the bootstrap distribution instead of assuming residual is normal. But rest of the assumptions is perfectly fine and if even the homoscedasticity does not hold good you can still work with paired bootstrap statistics and it will still work.

So, bootstrap regression is a big help for the statistician in case you have to do a statistical inference, that particularly if the model prediction is good. If the model prediction is not good then of course, you have to figure out a better model in that case.

But if the model prediction is good if the model is doing reasonable prediction in the out of the sample, if the model is doing if the RMSE if pretty low, if the R square adjusted R square is reasonably acceptable range for the given domain. Then just for inference you may find that the normality assumption is too strong assumption.

In that case you can do a statistical inference using bootstrap statistics and still you can go on with that prediction of the model, that predictive model because that predictive model do reason if it does do if it does good with prediction good prediction, then inference can be done using bootstrap sampling distribution. So, you will be good in that sense. You can salvage that situation when the model check assumption is not holding good.

As long as the IID sample assumption is working fine. You can salvage that situation using simple bootstrap statistic technique. So, with that I will stop here for this lecture and looking forward to the next lecture. I hope you enjoy this hands on session it is a very useful I find it in my many real life project I find it very useful this technique. So, I hope you will enjoy it and you will be able to use it and for your project experience.

So, thank you very much and see you in the next topic and next video next lecture.