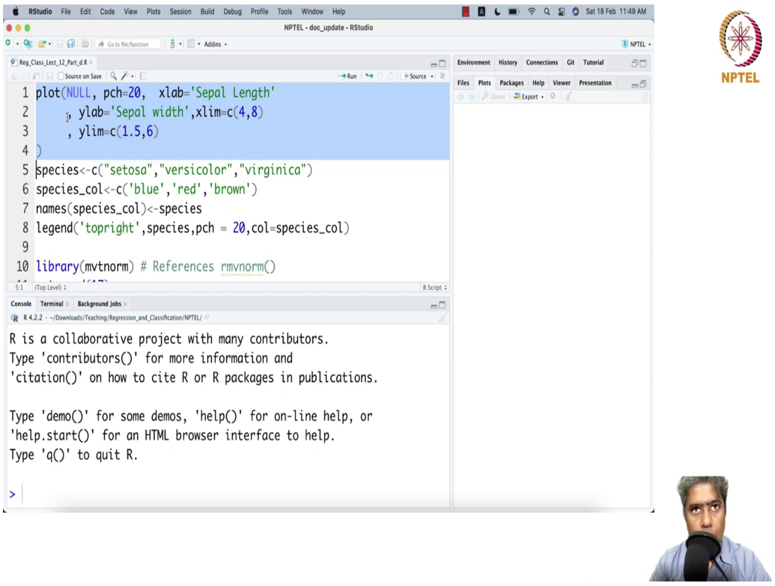
Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Chennai Mathematical Institute

Lecture - 43
Hands on with R: Implement LDA

Welcome back to the part d of lecture 12. In this video, we are going to do some Hands on.

(Refer Slide Time: 00:34)

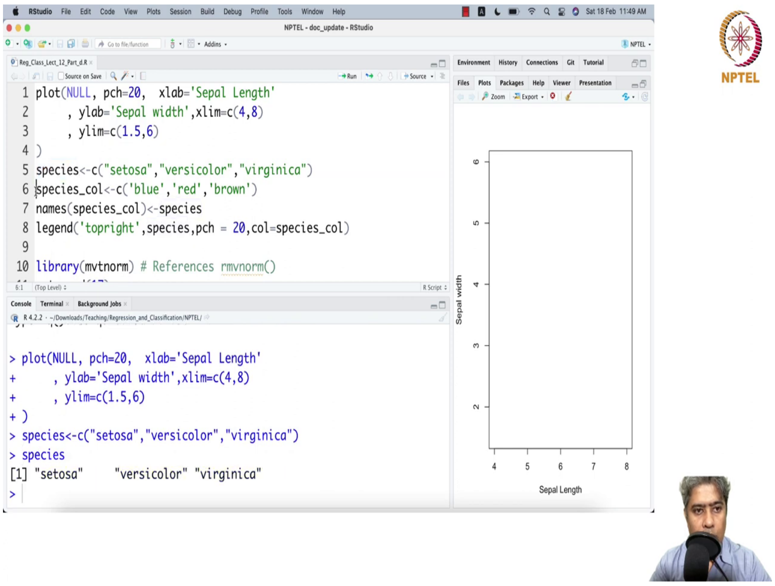


```r
plot(NULL, pch=20,  xlab='Sepal Length'
      , ylab='Sepal width',xlim=c(4,8)
      , ylim=c(1.5,6)
)
species<-c("setosa","versicolor","virginica")
species_col<-c('blue','red','brown')
names(species_col)<-species
legend('topright',species,pch = 20,col=species_col)

library(mvtnorm) # References rmvnorm()
```

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

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

>
```

So, let me start my R; in this. So, we will still we will work with the iris dataset. So, here I am going to first create a null plot. So, I we will just first see how the you know how the data looks like.

(Refer Slide Time: 00:59)

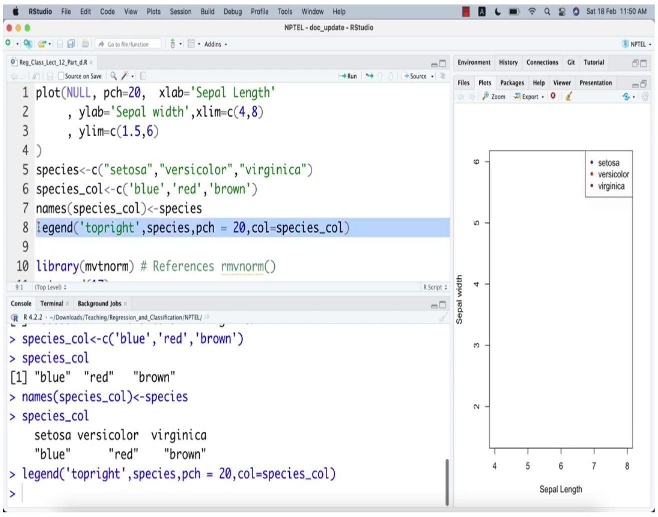


```
plot(NULL, pch=20,  xlab='Sepal Length'
      , ylab='Sepal width',xlim=c(4,8)
      , ylim=c(1.5,6)
)
species<-c("setosa","versicolor","virginica")
species_col<-c('blue','red','brown')
names(species_col)<-species
legend('topright',species,pch = 20,col=species_col)

library(mvtnorm) # References rmvnorm()
```

```
> plot(NULL, pch=20,  xlab='Sepal Length'
+       , ylab='Sepal width',xlim=c(4,8)
+       , ylim=c(1.5,6)
+ )
> species<-c("setosa","versicolor","virginica")
> species
[1] "setosa"     "versicolor" "virginica"
>
```

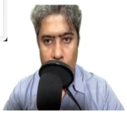
So, I am going to see the, on the x-axis, I am going to put sepal length. On the y-axis, I am going to put sepal width. So, I am going to name the species. There are 3 kind of species are there.

(Refer Slide Time: 01:18)



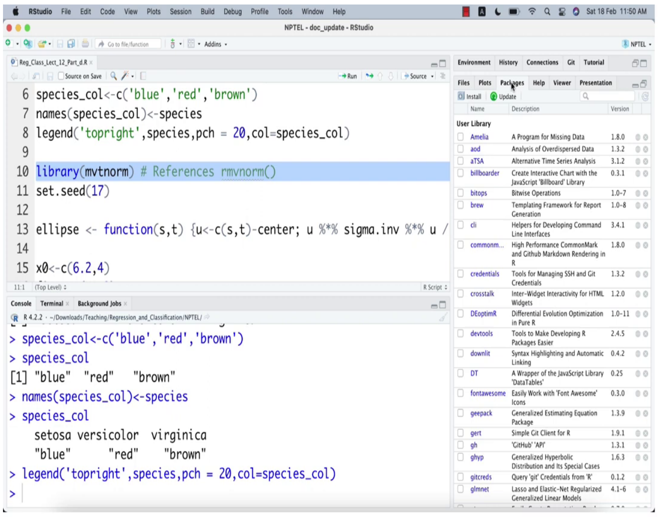
```
plot(NULL, pch=20,  xlab='Sepal Length'
     , ylab='Sepal width',xlim=c(4,8)
     , ylim=c(1.5,6)
)
species<-c("setosa","versicolor","virginica")
species_col<-c('blue','red','brown')
names(species_col)<-species
legend('topright',species,pch = 20,col=species_col)

library(mvtnorm) # References rmvnorm()
```

```
> species_col<-c('blue','red','brown')
> species_col
[1] "blue"  "red"   "brown"
> names(species_col)<-species
> species_col
    setosa versicolor  virginica
    "blue"      "red"    "brown"
> legend('topright',species,pch = 20,col=species_col)
>
```



And there species color I am going to assign blue, red and brown. And in the species color, I am also giving the name setosa got blue, versicolor got red and virginica got brown.

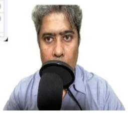
(Refer Slide Time: 01:45)


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Sat 18 Feb 11:50 AM
NPTEL - doc_update - RStudio
Go to file/function
Addins
NPTEL
Reg_Class_Lect_12_Part_d.R
Source on Save
Run
Source
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Install
Update
R Script
Console Terminal Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/

```r
species_col<-c('blue','red','brown')
names(species_col)<-species
legend('topright',species,pch = 20,col=species_col)

library(mvtnorm) # References rmvnorm()
set.seed(17)

ellipse <- function(s,t) {u<-c(s,t)-center; u %*% sigma.inv %*% u /

x0<-c(6.2,4)
```

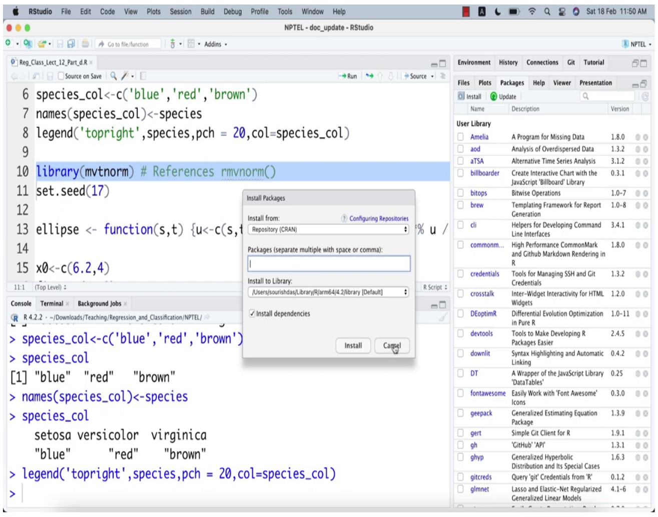
```
> species_col<-c('blue','red','brown')
> species_col
[1] "blue"  "red"   "brown"
> names(species_col)<-species
> species_col
    setosa versicolor  virginica
    "blue"      "red"    "brown"
> legend('topright',species,pch = 20,col=species_col)
>
```

	Name	Description	Version
	User Library		
□	Amelia	A Program for Missing Data	1.8.0
□	aod	Analysis of Overdispersed Data	1.3.2
□	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	Helpers for Developing Command Line Interfaces	3.4.1
□	commonm...	High Performance CommonMark and Github Markdown Rendering in R	1.8.0
□	credentials	Tools for Managing SSH and Git Credentials	1.3.2
□	crosstalk	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	Syntax Highlighting and Automatic Linking	0.4.2
□	DT	A Wrapper of the JavaScript Library 'DataTables'	0.25
□	fontawesome	Easily Work with 'Font Awesome' Icons	0.3.0
□	geepack	Generalized Estimating Equation Package	1.3.9
□	gert	Simple Git Client for R	1.9.1
□	gh	'GitHub' 'API'	1.3.1
□	ghyp	Generalized Hyperbolic Distribution and Its Special Cases	1.6.3
□	gitcreds	Query 'git' Credentials from 'R'	0.1.2
□	glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	4.1-6



And then, I am saying that legend put the legend on the top right. So, the setosa blue, versicolor red, virginica got brown. You can try some other color, no problem. Now, I am calling mvtnorm package. If you do not have the package, please install by, here you can go to the package.

(Refer Slide Time: 02:05)



```
species_col<-c('blue','red','brown')
names(species_col)<-species
legend('topright',species,pch = 20,col=species_col)

library(mvtnorm) # References rmvnorm()
set.seed(17)

ellipse <- function(s,t) {u<-c(s,t                          % u /

x0<-c(6.2,4)
```

```
> species_col<-c('blue','red','brown')
> species_col
[1] "blue"  "red"   "brown"
> names(species_col)<-species
> species_col
    setosa versicolor  virginica
    "blue"      "red"    "brown"
> legend('topright',species,pch = 20,col=species_col)
>
```

Install Packages

Install from: Repository (CRAN)

Packages (separate multiple with space or comma):

Install to Library: /Users/sourishdas/Library/R/arm64/4.2/library [Default]

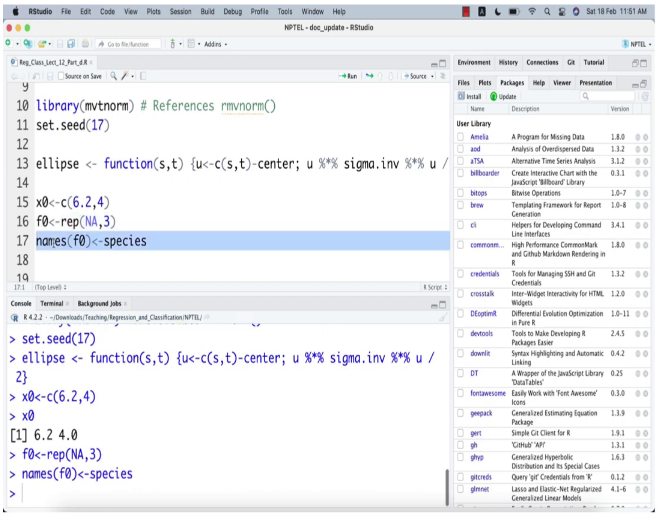
☑ Install dependencies

	Name	Description	Version
	User Library		
□	Amelia	A Program for Missing Data	1.8.0
□	aod	Analysis of Overdispersed Data	1.3.2
□	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	Helpers for Developing Command Line Interfaces	3.4.1
□	commonm...	High Performance CommonMark and Github Markdown Rendering in R	1.8.0
□	credentials	Tools for Managing SSH and Git Credentials	1.3.2
□	crosstalk	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	Syntax Highlighting and Automatic Linking	0.4.2
□	DT	A Wrapper of the JavaScript Library 'DataTables'	0.25
□	fontawesome	Easily Work with 'Font Awesome' Icons	0.3.0
□	geepack	Generalized Estimating Equation Package	1.3.9
□	gert	Simple Git Client for R	1.9.1
□	gh	'GitHub' 'API'	1.3.1
□	ghyp	Generalized Hyperbolic Distribution and Its Special Cases	1.6.3
□	gitcreds	Query 'git' Credentials from 'R'	0.1.2
□	glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	4.1-6

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Sat 18 Feb 11:50 AM
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R
Source on Save
Run
Source
R Script
Console Terminal Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Install Update
Configuring Repositories
Install
Cancel

Then install, and write this thing mvtnorm. Make sure your depend install dependencies corrected and then install. That will work.

(Refer Slide Time: 02:20)



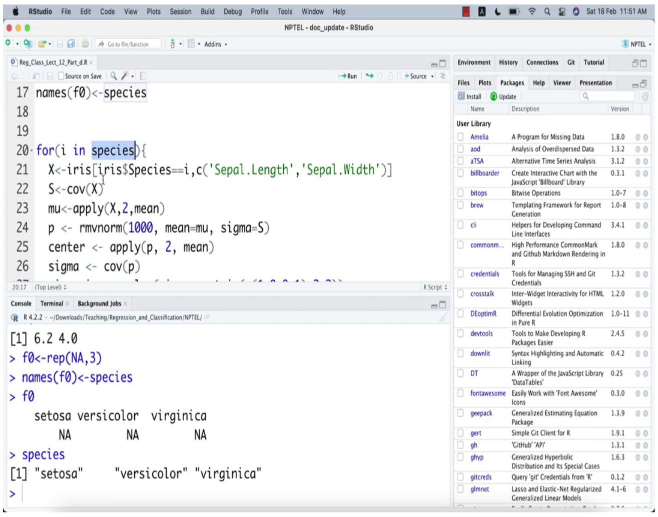
```
library(mvtnorm) # References rmvnorm()
set.seed(17)

ellipse <- function(s,t) {u<-c(s,t)-center; u %*% sigma.inv %*% u /

x0<-c(6.2,4)
f0<-rep(NA,3)
names(f0)<-species
```

```
> set.seed(17)
> ellipse <- function(s,t) {u<-c(s,t)-center; u %*% sigma.inv %*% u /
 2}
> x0<-c(6.2,4)
> x0
[1] 6.2 4.0
> f0<-rep(NA,3)
> names(f0)<-species
>
```

	Name	Description	Version
	User Library		
□	Amelia	A Program for Missing Data	1.8.0
□	aod	Analysis of Overdispersed Data	1.3.2
□	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	Helpers for Developing Command Line Interfaces	3.4.1
□	commonm...	High Performance CommonMark and Github Markdown Rendering in R	1.8.0
□	credentials	Tools for Managing SSH and Git Credentials	1.3.2
□	crosstalk	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	Syntax Highlighting and Automatic Linking	0.4.2
□	DT	A Wrapper of the JavaScript Library 'DataTables'	0.25
□	fontawesome	Easily Work with 'Font Awesome' Icons	0.3.0
□	geepack	Generalized Estimating Equation Package	1.3.9
□	gert	Simple Git Client for R	1.9.1
□	gh	'GitHub' 'API'	1.3.1
□	ghyp	Generalized Hyperbolic Distribution and Its Special Cases	1.6.3
□	gitcreds	Query 'git' Credentials from 'R'	0.1.2
□	glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	4.1-6

I am cancelling it because I already have it here. Setting a seed here. This function, one line function will help us to draw a ellipse or the contours of the Gaussian distribution. So, x naught is the point for its sort of the 6.2 and 4, is the point I am we are going to use as a test point. And then, I am going to create a f naught with the name species. So, f naught is going to be my likelihood or function.

(Refer Slide Time: 02:59)


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Sat 18 Feb 11:51 AM
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R
Source on Save
Run
Source
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Install
Update
Console Terminal Background Jobs
R Script
20:17 (Top Level)

```
names(f0)<-species


for(i in species){
  X<-iris[iris$Species==i,c('Sepal.Length','Sepal.Width')]
  S<-cov(X)
  mu<-apply(X,2,mean)
  p <- rmvnorm(1000, mean=mu, sigma=S)
  center <- apply(p, 2, mean)
  sigma <- cov(p)
```

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

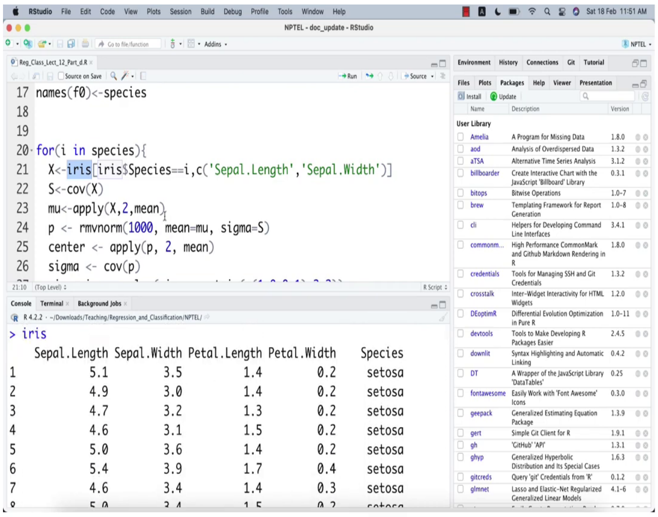
```
[1] 6.2 4.0
> f0<-rep(NA,3)
> names(f0)<-species
> f0
    setosa versicolor  virginica
        NA         NA         NA
> species
[1] "setosa"     "versicolor" "virginica"
>
```

User Library

	Name	Description	Version
□	Amelia	A Program for Missing Data	1.8.0
□	aod	Analysis of Overdispersed Data	1.3.2
□	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	Helpers for Developing Command Line Interfaces	3.4.1
□	commonm...	High Performance CommonMark and Github Markdown Rendering in R	1.8.0
□	credentials	Tools for Managing SSH and Git Credentials	1.3.2
□	crosstalk	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	Syntax Highlighting and Automatic Linking	0.4.2
□	DT	A Wrapper of the JavaScript Library 'DataTables'	0.25
□	fontawesome	Easily Work with 'Font Awesome' Icons	0.3.0
□	geepack	Generalized Estimating Equation Package	1.3.9
□	gert	Simple Git Client for R	1.9.1
□	gh	'GitHub' 'API'	1.3.1
□	ghyp	Generalized Hyperbolic Distribution and Its Special Cases	1.6.3
□	gitcreds	Query 'git' Credentials from 'R'	0.1.2
□	glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	4.1-6

So, i is in species. So, there are 3 species, for each species I am going to work with.

(Refer Slide Time: 03:14)



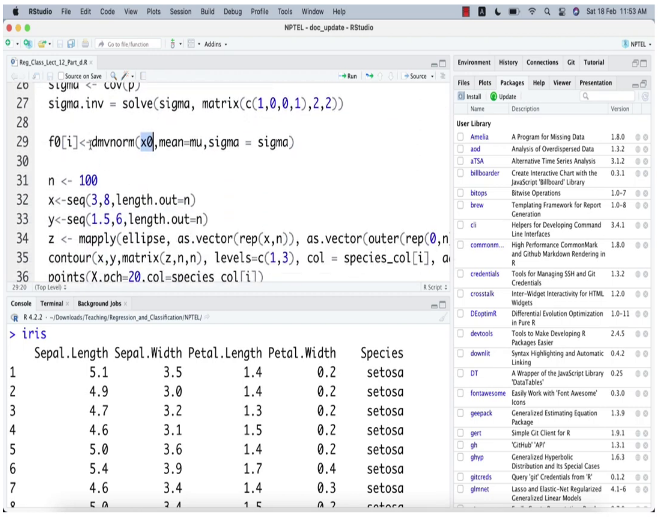
```r
names(f0)<-species


for(i in species){
  X<-iris[iris$Species==i,c('Sepal.Length','Sepal.Width')]
  S<-cov(X)
  mu<-apply(X,2,mean)
  p <- rmvnorm(1000, mean=mu, sigma=S)
  center <- apply(p, 2, mean)
  sigma <- cov(p)
```

```
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          5.1         3.5          1.4         0.2    setosa
2          4.9         3.0          1.4         0.2    setosa
3          4.7         3.2          1.3         0.2    setosa
4          4.6         3.1          1.5         0.2    setosa
5          5.0         3.6          1.4         0.2    setosa
6          5.4         3.9          1.7         0.4    setosa
7          4.6         3.4          1.4         0.3    setosa
```

Name	Description	Version
User Library		
Amelia	A Program for Missing Data	1.8.0
aod	Analysis of Overdispersed Data	1.3.2
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	Helpers for Developing Command Line Interfaces	3.4.1
commonm...	High Performance CommonMark and Github Markdown Rendering in R	1.8.0
credentials	Tools for Managing SSH and Git Credentials	1.3.2
crosstalk	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	Syntax Highlighting and Automatic Linking	0.4.2
DT	A Wrapper of the JavaScript Library 'DataTables'	0.25
fontawesome	Easily Work with 'Font Awesome' Icons	0.3.0
geepack	Generalized Estimating Equation Package	1.3.9
gert	Simple Git Client for R	1.9.1
gh	'GitHub' 'API'	1.3.1
ghyp	Generalized Hyperbolic Distribution and Its Special Cases	1.6.3
gitcreds	Query 'git' Credentials from 'R'	0.1.2
glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	4.1-6

From the iris data set, from the iris data set, what I am going to do. Whenever, iris species equal to the species name, then you just take the sepal length and sepal width that will be the x, ok. And then, you compute the covariance matrix of these two column and the mean of these two column.

(Refer Slide Time: 03:47)



```r
sigma <- cov(p)
sigma.inv = solve(sigma, matrix(c(1,0,0,1),2,2))

f0[i]<-dmvnorm(x0,mean=mu,sigma = sigma)

n <- 100
x<-seq(3,8,length.out=n)
y<-seq(1.5,6,length.out=n)
z <- mapply(ellipse, as.vector(rep(x,n)), as.vector(outer(rep(0,n
contour(x,y,matrix(z,n,n), levels=c(1,3), col = species_col[i], a
points(X,pch=20,col=species_col[i])
```

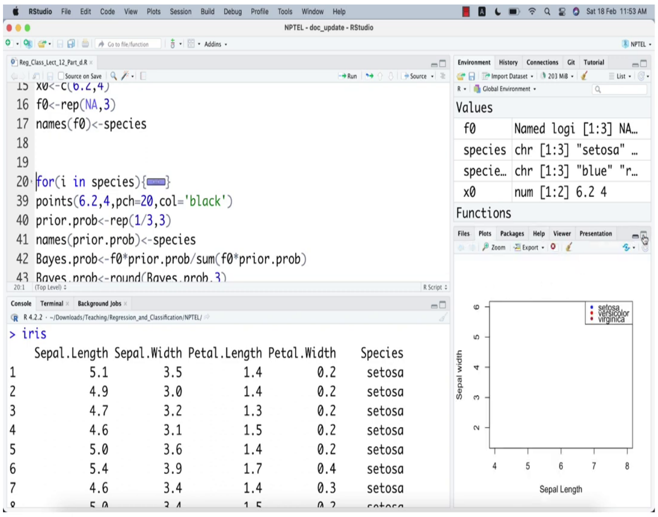
```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
1          5.1         3.5          1.4         0.2   setosa
2          4.9         3.0          1.4         0.2   setosa
3          4.7         3.2          1.3         0.2   setosa
4          4.6         3.1          1.5         0.2   setosa
5          5.0         3.6          1.4         0.2   setosa
6          5.4         3.9          1.7         0.4   setosa
7          4.6         3.4          1.4         0.3   setosa
8          5.0         3.4          1.5         0.2   setosa
```

	Name	Description	Version
User Library			
□	Amelia	A Program for Missing Data	1.8.0
□	aod	Analysis of Overdispersed Data	1.3.2
□	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	Helpers for Developing Command Line Interfaces	3.4.1
□	commonm...	High Performance CommonMark and Github Markdown Rendering in R	1.8.0
□	credentials	Tools for Managing SSH and Git Credentials	1.3.2
□	crosstalk	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	Syntax Highlighting and Automatic Linking	0.4.2
□	DT	A Wrapper of the JavaScript Library 'DataTables'	0.25
□	fontawesome	Easily Work with 'Font Awesome' Icons	0.3.0
□	geepack	Generalized Estimating Equation Package	1.3.9
□	gert	Simple Git Client for R	1.9.1
□	gh	'GitHub' 'API'	1.3.1
□	ghyp	Generalized Hyperbolic Distribution and Its Special Cases	1.6.3
□	gitcreds	Query 'git' Credentials from 'R'	0.1.2
□	glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models	4.1-6

Now, given this mean and this covariance matrix, you simulate 1000 points and compute the centre of the point and the covariance of the of those points, ok. And then, you calculate the sigma inverse. So, that gives you a; then you use this mu and sigma to calculate the in call that multivariate normal, density put it in the multi dmvnorm is the density of the multivariate normal and plug it in.

This is the likelihood value of the multivariate density for x naught, x naught point, x naught is the point at which we want to estimate. So, what is the likelihood for that point? Then, for n equal to 100, we creating x axis, y axis, and we are saying that, ok, please draw in this line we are 2, 3 lines we are drawing the contour for that particular species, ok.

(Refer Slide Time: 05:04)



```
x0<-c(6.2,4)
f0<-rep(NA,3)
names(f0)<-species


for(i in species){...}
points(6.2,4,pch=20,col='black')
prior.prob<-rep(1/3,3)
names(prior.prob)<-species
Bayes.prob<-f0*prior.prob/sum(f0*prior.prob)
Bayes.prob<-round(Bayes.prob,3)
```

```
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
1          5.1         3.5          1.4         0.2   setosa
2          4.9         3.0          1.4         0.2   setosa
3          4.7         3.2          1.3         0.2   setosa
4          4.6         3.1          1.5         0.2   setosa
5          5.0         3.6          1.4         0.2   setosa
6          5.4         3.9          1.7         0.4   setosa
7          4.6         3.4          1.4         0.3   setosa
8          5.0         3.4          1.5         0.2   setosa
```

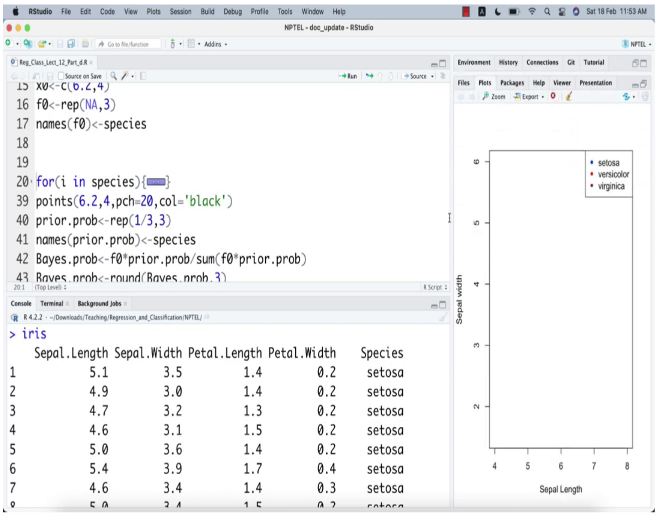
Values

```
f0       Named logi [1:3] NA…
species  chr [1:3] "setosa" …
specie…  chr [1:3] "blue" "r…
x0       num [1:2] 6.2 4
```

Functions

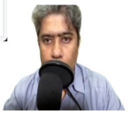
So, if let me just run this loop.

(Refer Slide Time: 05:13)



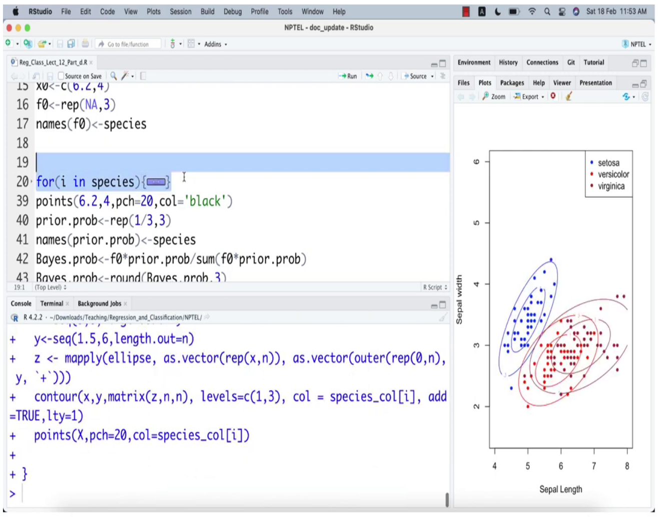
```
x0<-c(6.2,4)
f0<-rep(NA,3)
names(f0)<-species


for(i in species){...}
points(6.2,4,pch=20,col='black')
prior.prob<-rep(1/3,3)
names(prior.prob)<-species
Bayes.prob<-f0*prior.prob/sum(f0*prior.prob)
Bayes.prob<-round(Bayes.prob,3)
```

```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
```



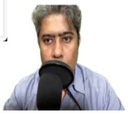
So, here is the viewer, plots. So, here is the plots.

(Refer Slide Time: 05:17)



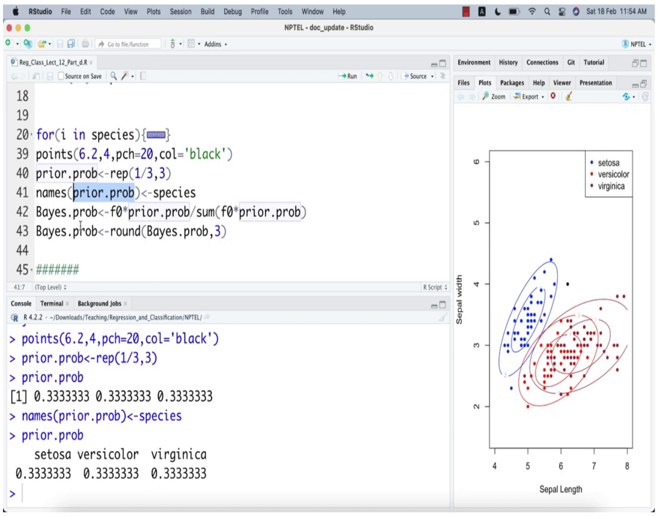
```
x0<-c(6.2,4)
f0<-rep(NA,3)
names(f0)<-species

for(i in species){
points(6.2,4,pch=20,col='black')
prior.prob<-rep(1/3,3)
names(prior.prob)<-species
Bayes.prob<-f0*prior.prob/sum(f0*prior.prob)
Bayes.prob<-round(Bayes.prob,3)
```

```
+   y<-seq(1.5,6,length.out=n)
+   z <- mapply(ellipse, as.vector(rep(x,n)), as.vector(outer(rep(0,n),
 y, `+`)))
+   contour(x,y,matrix(z,n,n), levels=c(1,3), col = species_col[i], add
=TRUE,lty=1)
+   points(X,pch=20,col=species_col[i])
+
+ }
>
```



So, let me just run the points. So, these are the 3 points and 3.

(Refer Slide Time: 05:28)



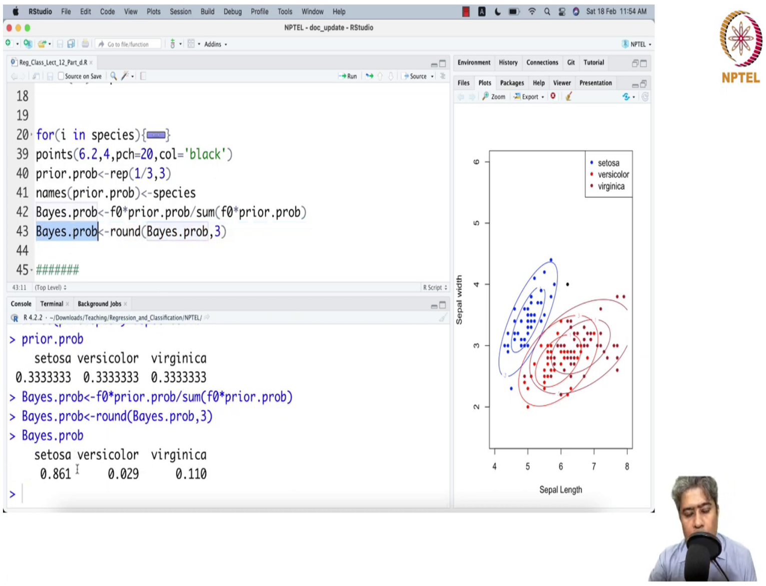
```
for(i in species){…}
points(6.2,4,pch=20,col='black')
prior.prob<-rep(1/3,3)
names(prior.prob)<-species
Bayes.prob<-f0*prior.prob/sum(f0*prior.prob)
Bayes.prob<-round(Bayes.prob,3)

#######
```

```
> points(6.2,4,pch=20,col='black')
> prior.prob<-rep(1/3,3)
> prior.prob
[1] 0.3333333 0.3333333 0.3333333
> names(prior.prob)<-species
> prior.prob
    setosa versicolor  virginica 
 0.3333333  0.3333333  0.3333333 
> 
```

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R
Source on Save
Run
Source
41:7
(Top Level)
R Script
Console
Terminal
Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Zoom
Export
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8

And then, here is the test point, here is the test point. You see this is the test point. Now, I am going to give equal prior probability. I am not saying, so prior probability for this point belong to setosa, versicolor or virginica is equal. I am not giving any special additional preference to any point. So, the name prior of the, now I just add the names also, and then, I am just computing the Bayes probability, ok.

(Refer Slide Time: 06:03)



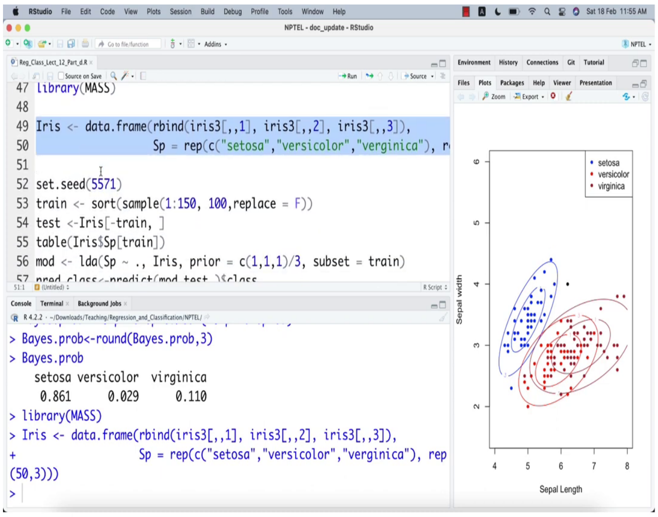
```
for(i in species){…}
points(6.2,4,pch=20,col='black')
prior.prob<-rep(1/3,3)
names(prior.prob)<-species
Bayes.prob<-f0*prior.prob/sum(f0*prior.prob)
Bayes.prob<-round(Bayes.prob,3)

#######
```

```
> prior.prob
   setosa versicolor  virginica
0.3333333  0.3333333  0.3333333
> Bayes.prob<-f0*prior.prob/sum(f0*prior.prob)
> Bayes.prob<-round(Bayes.prob,3)
> Bayes.prob
   setosa versicolor  virginica
    0.861      0.029      0.110
>
```

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R
Console Terminal Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8
NPTEL

And rounding of the Bayes probability up to the 3 decimal process, so here. So, that means, Bayes probability is 0.861. This point belongs to setosa with 80 percent probability to versicolor with 0 point 2.9 percent or about 3 percent probability and 11percent probability that it belongs to virginica. So, with a high probability we will say that most likely this point belongs to setosa.

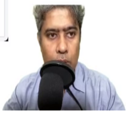
(Refer Slide Time: 06:40)



RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R
Source on Save
Run
Source
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Zoom Export
Console Terminal Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8

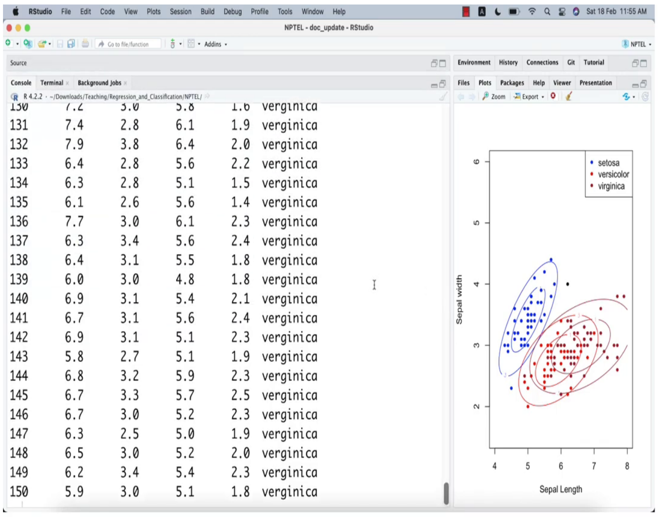
```
library(MASS)

Iris <- data.frame(rbind(iris3[,,1], iris3[,,2], iris3[,,3]),
                   Sp = rep(c("setosa","versicolor","verginica"), r

set.seed(5571)
train <- sort(sample(1:150, 100,replace = F))
test <-Iris[-train, ]
table(Iris$Sp[train])
mod <- lda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
pred.class<-predict(mod,test )$class
```

```
> Bayes.prob<-round(Bayes.prob,3)
> Bayes.prob
   setosa versicolor  virginica
    0.861      0.029      0.110
> library(MASS)
> Iris <- data.frame(rbind(iris3[,,1], iris3[,,2], iris3[,,3]),
+                   Sp = rep(c("setosa","versicolor","verginica"), rep
(50,3)))
> 
```

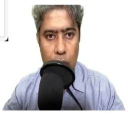


Now, we are going to do the, this is for one test point. So, I try to demonstrate how linear discriminant analysis in theory, using one test point. But, most of the time you will see that your data will have many test points and you have to run it like a typical ml setup, machine learning setup.

(Refer Slide Time: 07:16)

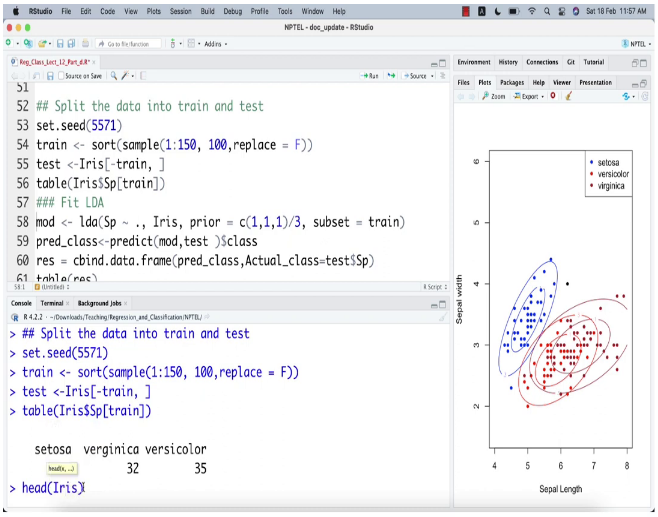


```
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
130       7.2       3.0       5.8       1.6   verginica
131       7.4       2.8       6.1       1.9   verginica
132       7.9       3.8       6.4       2.0   verginica
133       6.4       2.8       5.6       2.2   verginica
134       6.3       2.8       5.1       1.5   verginica
135       6.1       2.6       5.6       1.4   verginica
136       7.7       3.0       6.1       2.3   verginica
137       6.3       3.4       5.6       2.4   verginica
138       6.4       3.1       5.5       1.8   verginica
139       6.0       3.0       4.8       1.8   verginica
140       6.9       3.1       5.4       2.1   verginica
141       6.7       3.1       5.6       2.4   verginica
142       6.9       3.1       5.1       2.3   verginica
143       5.8       2.7       5.1       1.9   verginica
144       6.8       3.2       5.9       2.3   verginica
145       6.7       3.3       5.7       2.5   verginica
146       6.7       3.0       5.2       2.3   verginica
147       6.3       2.5       5.0       1.9   verginica
148       6.5       3.0       5.2       2.0   verginica
149       6.2       3.4       5.4       2.3   verginica
150       5.9       3.0       5.1       1.8   verginica
```



So, I am going to take the iris dataset with; here is the iris dataset with all the 150 values, with 50 in each probability. And then, I am going to first Split the data into train and test, ok. Split the data into train and test, ok.

(Refer Slide Time: 07:37)



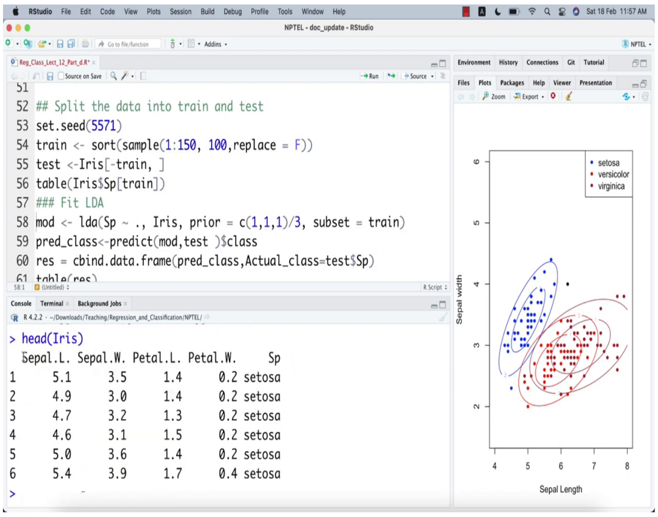
RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R*
Source on Save
Run
Source
Console
Terminal
Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Zoom
Export
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8

```
## Split the data into train and test
set.seed(5571)
train <- sort(sample(1:150, 100,replace = F))
test <-Iris[-train, ]
table(Iris$Sp[train])
### Fit LDA
mod <- lda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
pred_class<-predict(mod,test )$class
res = cbind.data.frame(pred_class,Actual_class=test$Sp)
table(res)
```

```
> ## Split the data into train and test
> set.seed(5571)
> train <- sort(sample(1:150, 100,replace = F))
> test <-Iris[-train, ]
> table(Iris$Sp[train])

   setosa  verginica versicolor
                  32          35
> head(Iris)
```

So, I am just, I just played the data. Till the data table if you just say; so, basically in train data there were 33 setosa, 32 virginica and 35 were versicolor. So, they are almost same. So, out of 150 samples, I have taken 100 random samples from the main data and those samples belongs to train datasets. 100 samples belongs to and 50 samples belong to the test dataset. Here in this, I am going to fit LDA, linear discriminant analysis. And I am saying that ok fit all the models.

(Refer Slide Time: 08:50)

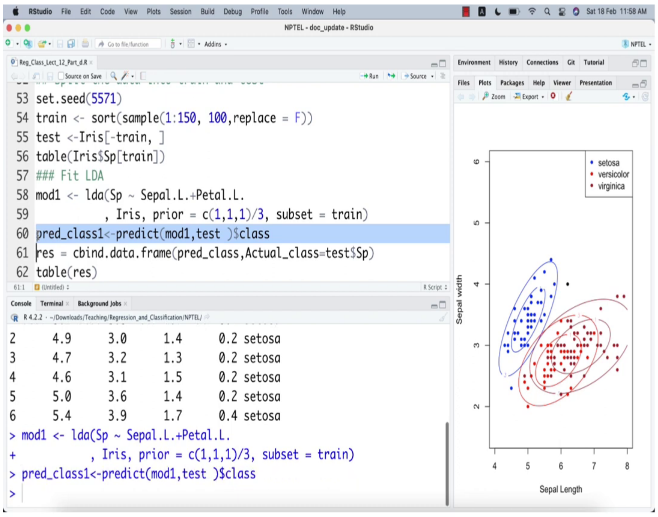


```r
## Split the data into train and test
set.seed(5571)
train <- sort(sample(1:150, 100,replace = F))
test <-Iris[-train, ]
table(Iris$Sp[train])
### Fit LDA
mod <- lda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
pred_class<-predict(mod,test )$class
res = cbind.data.frame(pred_class,Actual_class=test$Sp)
table(res)
```

```
> head(Iris)
 Sepal.L. Sepal.W. Petal.L. Petal.W.     Sp
1      5.1      3.5      1.4      0.2 setosa
2      4.9      3.0      1.4      0.2 setosa
3      4.7      3.2      1.3      0.2 setosa
4      4.6      3.1      1.5      0.2 setosa
5      5.0      3.6      1.4      0.2 setosa
6      5.4      3.9      1.7      0.4 setosa
>
```

So, maybe I will just say head Iris.

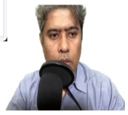
(Refer Slide Time: 08:55)



```
set.seed(5571)
train <- sort(sample(1:150, 100,replace = F))
test <-Iris[-train, ]
table(Iris$Sp[train])
### Fit LDA
mod1 <- lda(Sp ~ Sepal.L.+Petal.L.
            , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
res = cbind.data.frame(pred_class,Actual_class=test$Sp)
table(res)
```

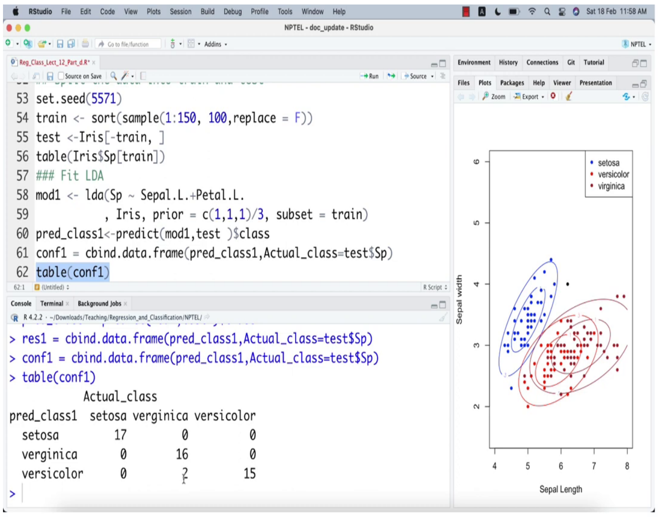
```
2      4.9      3.0      1.4      0.2 setosa
3      4.7      3.2      1.3      0.2 setosa
4      4.6      3.1      1.5      0.2 setosa
5      5.0      3.6      1.4      0.2 setosa
6      5.4      3.9      1.7      0.4 setosa
> mod1 <- lda(Sp ~ Sepal.L.+Petal.L.
+             , Iris, prior = c(1,1,1)/3, subset = train)
> pred_class1<-predict(mod1,test )$class
>
```

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R
Source on Save
Run
Source
Console
Terminal
Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Zoom
Export
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8



So, maybe first I will fit with length plus sepal width, maybe with petal length, ok with two feature, essentially with two feature. This is my first model. And if we just run this then we can do the same with the predictive class.

(Refer Slide Time: 09:28)



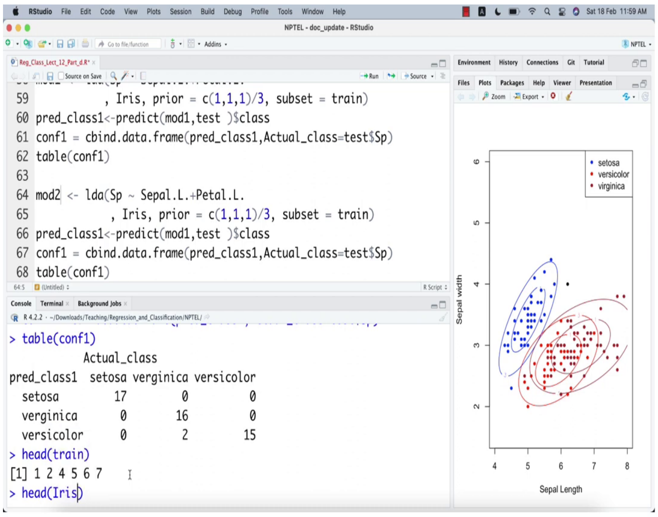
```
set.seed(5571)
train <- sort(sample(1:150, 100,replace = F))
test <-Iris[-train, ]
table(Iris$Sp[train])
### Fit LDA
mod1 <- lda(Sp ~ Sepal.L.+Petal.L.
            , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)
```

```
> res1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
> conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
> table(conf1)
            Actual_class
pred_class1  setosa verginica versicolor
  setosa         17         0          0
  verginica       0        16          0
  versicolor      0         2         15
>
```

RStudio
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R*
Console
Terminal
Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
setosa
versicolor
virginica
Sepal width
Sepal Length
6
5
4
3
2
4
5
6
7
8

And then this is the result 1. And this is the confusion matrix. Essentially, actually this is only the confusion matrix. So, I would rather instead of result I would say it is a confusion matrix. And let me just show you how the confusion matrix looks like, ok. There are two cases where virginica; which were virginica that was classified as versicolor, ok. So, this is the linear discriminant analysis with two feature.

(Refer Slide Time: 10:25)

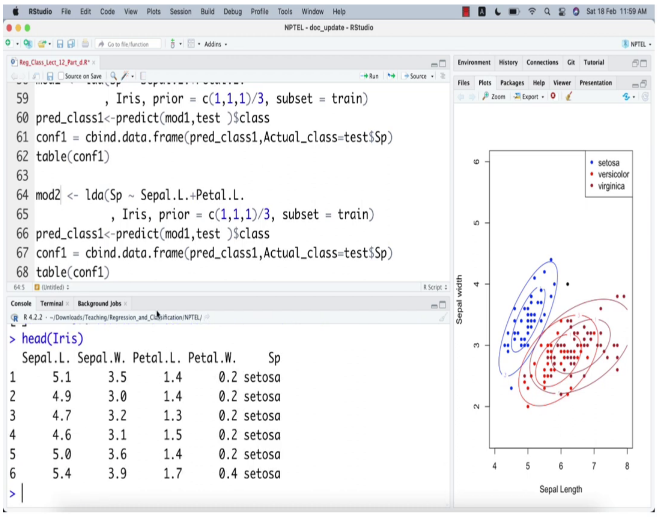

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
NPTEL - doc_update - RStudio
Reg_Class_Lect_12_Part_d.R*
Run
Source
Console
Terminal
Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8

```
              , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)

mod2 <- lda(Sp ~ Sepal.L.+Petal.L.
               , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)
```

```
> table(conf1)
            Actual_class
pred_class1  setosa verginica versicolor
  setosa         17         0          0
  verginica       0        16          0
  versicolor      0         2         15
> head(train)
[1] 1 2 4 5 6 7
> head(Iris)
```

Now, let me run the entire thing with maybe model 2. And what we can do; head train, I am sorry; Iris.

(Refer Slide Time: 10:41)

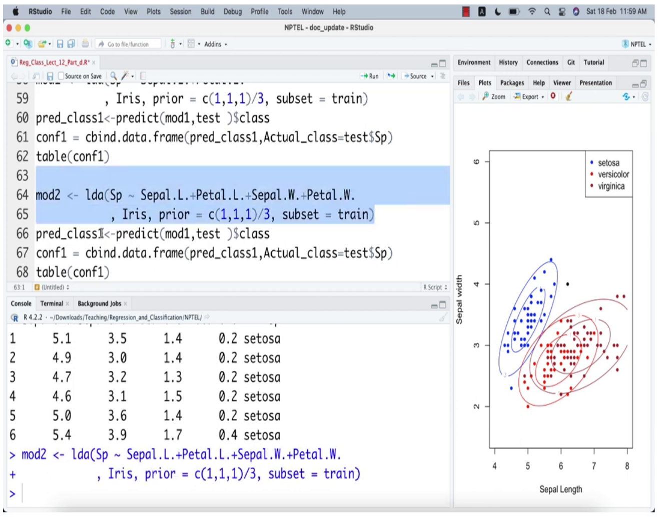


RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help
Sat 18 Feb 11:59 AM
NPTEL - doc_update - RStudio
Go to file/function
Addins
NPTEL
Reg_Class_Lect_12_Part_d.R*
Source on Save
Run
Source
Environment History Connections Git Tutorial
Files Plots Packages Help Viewer Presentation
Zoom
Export
64:5
(Untitled)
R Script
Console Terminal Background Jobs
R 4.2.2 · ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
setosa
versicolor
virginica
Sepal width
Sepal Length
2
3
4
5
6
4
5
6
7
8

```
            , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)

mod2 <- lda(Sp ~ Sepal.L.+Petal.L.
             , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)
```

```
> head(Iris)
  Sepal.L. Sepal.W. Petal.L. Petal.W.     Sp
1      5.1      3.5      1.4      0.2 setosa
2      4.9      3.0      1.4      0.2 setosa
3      4.7      3.2      1.3      0.2 setosa
4      4.6      3.1      1.5      0.2 setosa
5      5.0      3.6      1.4      0.2 setosa
6      5.4      3.9      1.7      0.4 setosa
> 
```

(Refer Slide Time: 10:48)

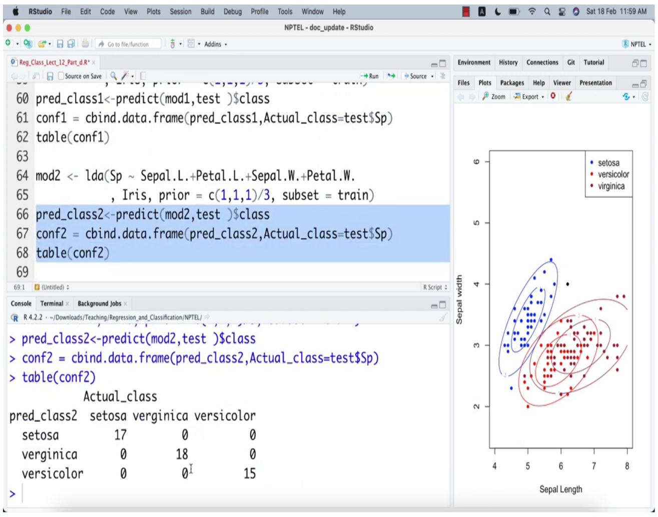


```
            , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)

mod2 <- lda(Sp ~ Sepal.L.+Petal.L.+Sepal.W.+Petal.W.
             , Iris, prior = c(1,1,1)/3, subset = train)
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)
```

```
1      5.1      3.5      1.4      0.2 setosa
2      4.9      3.0      1.4      0.2 setosa
3      4.7      3.2      1.3      0.2 setosa
4      4.6      3.1      1.5      0.2 setosa
5      5.0      3.6      1.4      0.2 setosa
6      5.4      3.9      1.7      0.4 setosa
> mod2 <- lda(Sp ~ Sepal.L.+Petal.L.+Sepal.W.+Petal.W.
+             , Iris, prior = c(1,1,1)/3, subset = train)
>
```

So, let me now just put the Sepal width and Petal width as well. So, all the features, all 4 features let me put, all the 4 features.

(Refer Slide Time: 11:00)



```
pred_class1<-predict(mod1,test )$class
conf1 = cbind.data.frame(pred_class1,Actual_class=test$Sp)
table(conf1)

mod2 <- lda(Sp ~ Sepal.L.+Petal.L.+Sepal.W.+Petal.W.
            , Iris, prior = c(1,1,1)/3, subset = train)
pred_class2<-predict(mod2,test )$class
conf2 = cbind.data.frame(pred_class2,Actual_class=test$Sp)
table(conf2)
```

```
> pred_class2<-predict(mod2,test )$class
> conf2 = cbind.data.frame(pred_class2,Actual_class=test$Sp)
> table(conf2)
            Actual_class
pred_class2  setosa verginica versicolor
  setosa         17         0          0
  verginica       0        18          0
  versicolor      0         0         15
>
```

And let me do the prediction. And out of the sample prediction, remember that here is my test data set to do the prediction, ok. And then this is the confusion matrix. So, let us run this confusion matrix, now it is being corrected. So, when we put the more feature, naturally the classification becomes better. So, and it is in the out of the sample classification it got better.

So, this shows that if you have more feature, more typically it helps you. But at the same time, you have to be careful about the over fitting that if you put too many features too many engineered features, it might, you might end up in a wrong place with lot of over fitting. That means in your test accuracy and out of the sample accuracy. And in sample accuracy will be very different.

At the same time, remember that you have to be this particular data set iris data set is actually a very toy data set. Going forward we will try with more real life new days data set, new age data set where it will be much much difficult to classify the target variable. With this, thank you very much. Wish you a happy weekend. See you in the next week with new lecture, new topic.

Thank you very much.