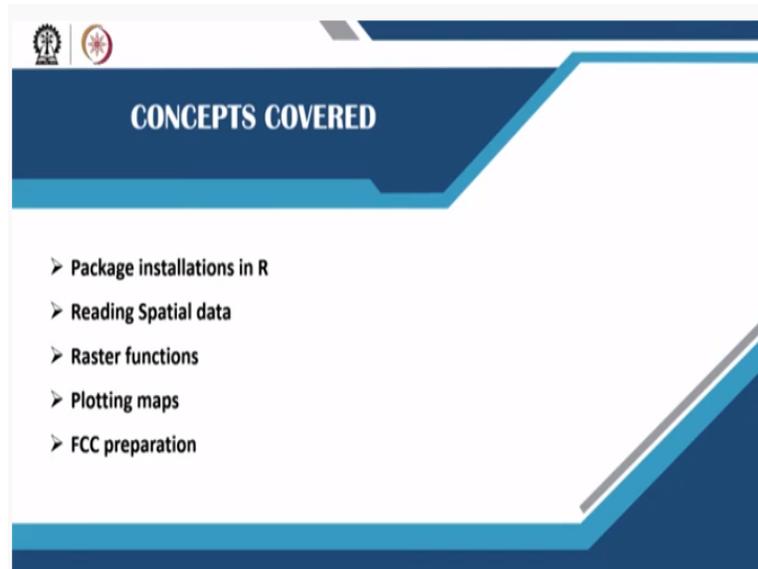


Geographic Information Systems
Mr. Prakash PS
Ranbir and Chitra Gupta School of Infrastructure Design and Management
Indian Institute of Technology – Kharagpur

Lecture – 56
Introduction to R (Part – II)

Welcome to the course on geographic information system. So this is module 11 GIS software and in this hands on session, we are going to look at introduction to R and its functionalities.

(Refer Slide Time: 00:36)



So the concept covered in this session are installing different packages in R reading spatial data and reading a raster and vector data's and different raster functions plotting maps in R and composing a false color composite using R. So in the previous session we have seen how to import excel files and how to subset the data and in in the upcoming sessions and this session we are going to see how in how to install different packages.

(Refer Slide Time: 01:07)

Package installation

R has several ready-made packages to perform various functions. A new package installation can be made with the function `install.packages`. It needs to be loaded to every R session with the `library` function.

Examples:

```
> install.packages("Utils")
> install.packages("rgdal")
> install.packages("raster")
> library(RStoolbox)

> library(utils)
> library(raster)
> library(rgdal)
> library(RStoolbox)
```

So packages can be installed using a command called install dot packages. So for this particular session we are going to require 3 important packages one is rgdal second is raster and third one is Rstoolbox. So once we install these packages, we also going to need call this packages as a library in our whenever we start R programming.

(Refer Slide Time: 01:44)

```
1 install.packages("utils")
2 install.packages("raster")
3 install.packages("rgdal")
4
5 library(utils)
6 library(raster)
7 library(rgdal)
8
9 setwd("C:/Users/dr. Bharath H Aithal/Desktop/NPTEL/Datasets_for_NPTEL/R sess
10
11
12 #read a raster data
13 image1 = brick(file.choose())
14 image2 = raster(file.choose())
15
16
17 #read a vector data
18 ani = readOVC(file.choose())
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

So let us jump to R program and startup the functions. So as I mentioned in the slide, so we need to install the packages here using install dot packages. So this is a onetime task for example if you want to install utils package, you just need to type install dot packages and run. So if you run install dot packages the utils package will get installed. So if it is already installed it will give

you an option saying that it is already installed do you want to reinstall again, or it will give some kind of warning if the version is not compatible.

So let us try to install next package which is raster. So we need to use double quotation inside of brackets to run this particular command. So once to once you try to install the raster it will get installed in your system. So to install these packages you need to have internet connection. So let us try to install rgdal package which is useful in bringing vector data which is useful in bringing vector data into the R environment it will take some time because of the package file. So the zip file will get downloaded to R environment and it will get installed by itself.

So once these files will get downloaded the next button is visible in the console. So that means the installation is done. So now we will try to call this one as a library here we need to call all these packages as a libraries in R so we can select all these together all these things and run it. So the libraries will get loaded in your systems. Once this is done let us try to set the directory for our file as I mentioned in the earlier session. So we need to use a forward slashes and using setwd function we can run we can set the directory for our working directory.

And to import the raster data we can use 2 command we can either use brick or raster. So similarly so instead of brick we can use image 2 is equal to raster and file dot choose function. So what is the difference between the brick and raster function. So brick is to import raster files having multiple bands. So raster is to bring the raster files having single bands.

Let us try to import a brick import a raster file using a brick command. I will run this using control and enter button. So once you do that the particular folder which we set as a directory will get open up. So if you go to import and open a raster file Jaipur quikbird PAN.

(Refer Slide Time: 04:45)

```
9 setwd("C:/Users/Dr. Bharath H Aithal/Desktop/NPTEL/Datasets_for_NPTEL/R session NPTEL/Data/Import/Jaipur_quikbird_PAN.tif")
10
11
12 #read a raster data
13 image1 = brick("C:/Users/Dr. Bharath H Aithal/Desktop/NPTEL/Datasets_for_NPTEL/R session NPTEL/Data/Import/Jaipur_quikbird_PAN.tif")
14 image2 = raster(file.choose())
15 |
16
17 #read a vector data
18 <
19
```

```
dimensions : 686, 599, 410914, 1 (nrow, ncol, ncell, nlayers)
resolution : 0.6, 0.6 (x, y)
extent      : 582122.4, 582481.8, 2976694, 2977105 (xmin, xmax, ymin, ymax)
crs        : +proj=utm +zone=43 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0
source     : C:/Users/Dr. Bharath H Aithal/Desktop/NPTEL/Datasets_for_NPTEL/R session NPTEL/Data/Import/Jaipur_quikbird_PAN.tif
names      : Jaipur_quikbird_PAN
min values : 0
max values : 65535
```

```
> #read a raster data
> image1 = brick("C:/Users/Dr. Bharath H Aithal/Desktop/NPTEL/Datasets_for_NPTEL/R session NPTEL/Data/Import/Jaipur_quikbird_PAN.tif")
>
```

So this particular image will get open. So we can see its properties by going to image 1 and typing it in console as image 1 and run it, it will show its properties. What are the properties of this particular raster? This raster is nothing but a special image having a resolution of 0.6 meter which is having 1 band having number of rows is 686, number of column is 599 and total number of cells and number of bands is 1 that means it is a single band raster image having special resolution of 0.6/0.6 and it also gives extent the left, right, top and bottom latitude and longitude are x minimum, x maximum, y minimum or y maximum.

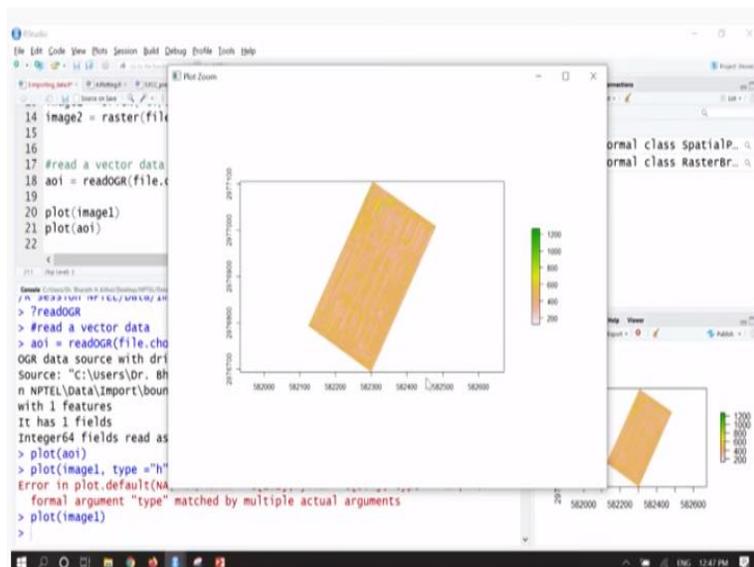
Also it used to give this information such as projection system is utm, zone is 43 and datum is wgs84 along with this it used the source from which directory this particular raster is coming from and what is its name? The file name and minimum and maximum value of the raster. So the entire metadata of the information or details of the information is visible once you run this particular image 1 variable so how do we get this image 1 variable by bringing in the raster using brick command.

So same process we can do using raster command as well just the difference is using brick command, we can bring multiple band raster's using raster we can bring in single band raster. So here inside brick there is a one more function there are 2 functions embedded here one is brick bracket and file dot choose bracket double quotations or double brackets. So if I remove this file dot choose and I can give the path here as well.

So the path of the file I can mention here and run this particular command. For example, I will show you show that instead of using a sub directory but instead of using file dot choose I can use the path of the file to import I just copy the path and put it inside brick data inside data there is a folder called import inside that there is a file called Jaipur underscore inside import folder there is a file called Jaipur underscore quik word underscore pan.

And just copy the name of this copy the name and bring it into qj bring it into R environment inside import folder there is a file called Jaipur quik PAN dot tif. So this is a tif file and then run it, so this also functions as importing a file. So instead of file dot choose we can do the path as well. So there is still we can input a raster data in R.

(Refer Slide Time: 08:35)



So let us try to input a vector data in R using a command called readOGR. So this particular command is coming from rgdal library. So how to check that suggest copy the function name and we can use question mark readOGR and run it right side we can see under help tab we can see readOGR coming from rgdal, readOGR vector maps into spatial object. So this is how we can see the help. Let us try to import an area of interest or using readOGR file.

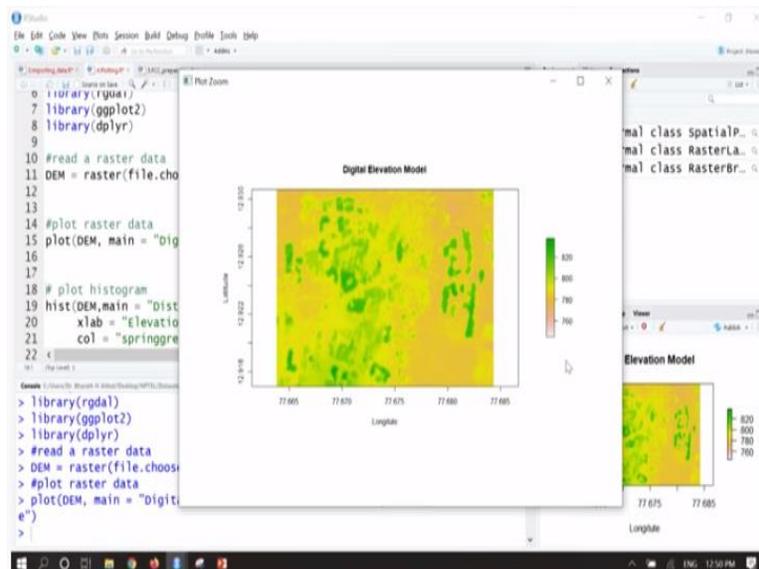
I will just run this particular command. So if you run this particular command so the window is not appearing but actually there is a red button which came in here. That means a window is behind the R, so this is here. So I will bring in the boundary this is aoi with which we are we are

bringing into the R environment currently the aoi is come here. Let us try to plot the aoi using plot aoi this is a vector file which is plotted in R.

So here you can see a box a rectangular box which is a share file we bought it into R environment. Similarly, we can plot Jaipur share file which we read using image 1 variable. So we can use a plot image 1 type is h. So either you can use a directly plot image one command to plot the raster. So by running this particular command we can plot the raster here in R we can zoom and see the plotted raster as well.

This is how we can import a different and the vector files into R environment. So if it is having multiple band 3 images will get plotted here. So the label also has come up the values so of the pixel values from 200 to 1200 also X and Y coordinate as well.

(Refer Slide Time: 10:53)



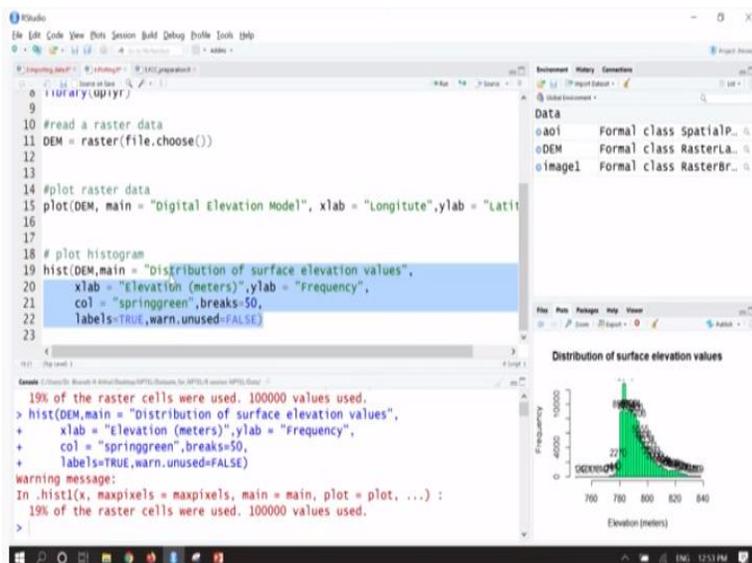
So for in the next exercise we will try to plot a digital elevation model using the same plot command and try to plot an histogram. So how to do that so to do that we need to have a library called ggplot and dplyr. So let us call these libraries at once select all this library and run. So here in the console you can see all the libraries will get loaded I will run it again. All the libraries have run so the setting working directory already done so the previous page we have done setting working directory it will open the same page.

So let us try to read a raster now. So let us try to read a digital elevation model since the red button is here so we can open the window. So inside a folder called data there is a digital elevation model file I will import digital elevation model into R environment so that has come up here we can see now we will try to plot digital elevation using plot command. So here under plot we need to mention the variable name digital elevation model and the main is to give the title and x lab and y lab used to give the x and y direction labels.

So if I click on the line 15 which is having the plot function and run it the digital elevation model will get plotted. So right now we have plotted digital elevation model here we can clearly see there is a title saying that digital elevation model which we have mentioned here, and x lab and y lab also had has come up latitude and longitude and latitude. So clearly mentioning it is somewhere in between 77 degrees and 12 degrees.

And at the right side, we can see the legend showing that elevation varying from 760 to 820 having varied colors. So the plotted map can be exported into a PDF or image format using export command. Here if you click on image it will get exported into image format if you want to convert it into PDF that also can be done.

(Refer Slide Time: 13:27)



So if you want to plot a histogram so histogram is a representative is the plot between the pixel values and the number of or the frequency of pixel values. So using hist command we can plot

the histogram. So here digital elevation model is the raster we need to call. So here we already have a digital elevation model and we can view the title as distribution of surface elevation values x lab is elevation y lab is the frequency.

Since we are using a histogram y lab will be the frequency and we can give number of breaks and the colors and labels are true. So if you run this entire command the histogram of this particular digital elevation will get plotted. So I will zoom, and we will inspect the histogram. So here we can see the elevation having more than somewhere around 710 meters is having maximum number of pixels.

Frequency versus elevation or pixel values is the histogram of this particular raster. This is how we can plot the histogram of any raster. So if you want to plot a histogram of an image also that also can be done. So instead of the elevation there will be pixel values in the x direction. So here also the variable such as brakes can be changed if you want to have more breaks or less breaks that also can be done.

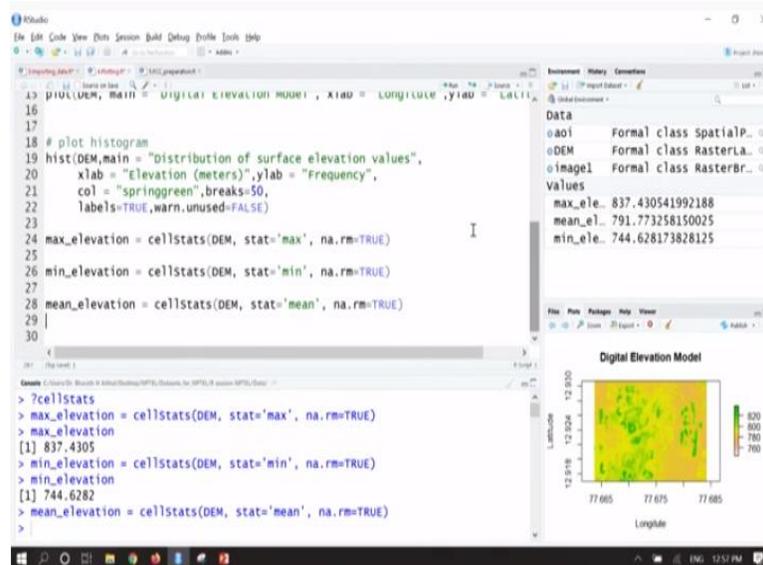
For example, if the breaks is 10 the histograms will have only 10 breaks. We will try to inspect that if you run that the number of breaks in histograms have changed to 10. Only 10 breaks are that if you want to have more finer breaks let us try to see that as well let us have 50 breaks in the histogram and run it. We need to select the entire command and then run it. So there are 50 breaks in the histogram.

So this is clearly a visible there are total number the number of breaks have been increased from 10 to 50. So in this exercise we have seen how to import the packages how to import the libraries and how to import a digital elevation model or any raster into R environment and finally plot it and to generate histogram of the raster. Next, we are going to see how to generate the raster statistics.

For example, if you want to see what is the maximum elevation value in this particular raster. For example, using this particular graph we will not be able to understand the what is the

maximum value of the pixel what is the mean value of the pixel? Those can be done using sales statistics command. We will try to see it here itself.

(Refer Slide Time: 16:30)



So there is a command called cell statistics There is a function called cells statistics We will try to see its syntax question mark cellstats and run it. So it shows that it gives the syntax for cells statistics. So here we just need to copy this syntax from the help command and put it into R environment. So this cells statistics can be of mean, maximum or minimum or standard deviation also.

So for this particular digital elevation model we will see what is the maximum elevation is equal to cellstats instead of x it will here it will be digital elevation model variable DEM I will type here DEM stat is going to be maximum this time and I remove all other variables which are not necessary and then run this particular command. So now we calculated what is the maximum elevation of this particular raster. So the value has come up with the right side we will also inspect here if you copy maximum elevation and put it here and run.

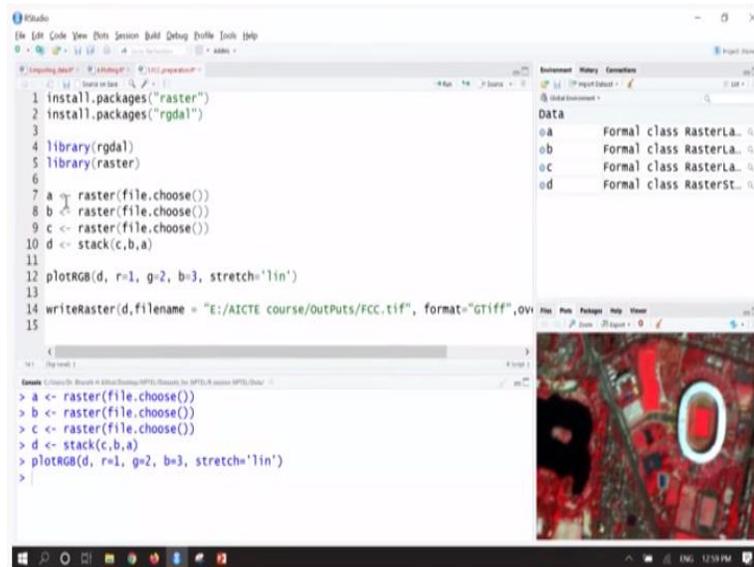
So it is saying 837.43, 837 is the maximum elevation of this particular raster which we are seeing in the screen left side. So in the plot we can see, so here it is showing 820. So what is the maximum value of the pixel in the entire scene? That is 837. These kinds of statistics, we can get it using cells statistics command. Let us also see what is the minimum value as well. So I will

copy the entire command and paste it into next line and instead of max I will say as min minimum and instead of maximum elevation it will be minimum elevation and then run it.

So we see what is the minimum elevation copy these things and put it into console and then run it. So minimum elevation is 747.62 the value ranges from 740 pixel value ranges from 744 to 837 this is how we can inspect the maximum and minimum value or after raster using cellstats command. We can also see mean and standard deviation as well. Let us try to inspect what is the mean value.

So if you use the same command and use mean as the stats and let us change the variable name as mean and then run it, it gives the mean value it is here as well. The mean value is 791.77 the calculations the raster calculations happen very fast this is one of the advantages of using R for simple calculation simple raster functions in R.

(Refer Slide Time: 19:54)



```
1 install.packages("raster")
2 install.packages("rgdal")
3
4 library(rgdal)
5 library(raster)
6
7 a <- raster(file.choose())
8 b <- raster(file.choose())
9 c <- raster(file.choose())
10 d <- stack(c,b,a)
11
12 plotRGB(d, r=1, g=2, b=3, stretch='lin')
13
14 writeRaster(d,filename = "E:/AICTE course/OUTPUTS/FCC.tif", format="GTiff",ovr
15
```

The screenshot shows the RStudio interface. The console window displays the execution of the R code. The Environment pane on the right shows four raster objects: 'a', 'b', 'c', and 'd', all of type 'Formal class RasterLa...'. The plot window shows a false color composite image of a landscape, with a prominent white circular feature in the center.

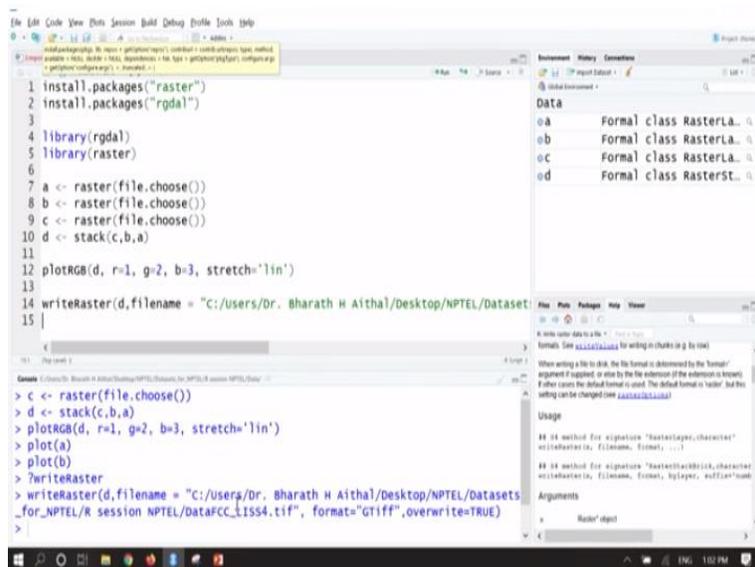
So let us proceed to the next exercise which is that is false color composite preparation. So in the earlier sessions we have seen how to download LISS4 or landsat imagery to your system using registering in Bhuvan website or USGS website and finally download into your system. In this session, we are going to see how to stack these images and prepare a false color composite. False color composite can be prepared using 3 bands.

So let us try to import the libraries as well Here I will import these 3 libraries and then path is already set since we have used the earlier R function as well. So I will clear the environment before doing the stacking I will clear the map view also can use the broom button to clear all these things. Once it is cleared let us try to import 3 bands, I will import the band a so if you run raster file dot choose and assign it to a so 1 band will get imported.

For false color composite, I have kept LISS band 2, LISS band 3 and LISS band 4. LISS band 2 is the green band and green red and infrared band. I will import the green band first. Once I import the green band next, I will import. Similarly, I import the red band as well and thirdly, I will import the infrared band also here infrared band I will import. So now we have 3 bands separately imported in R environment abc.

So simply we need to stack in opposite order cba using stat function. We can use stat function to make a stacking of these 3 images to prepare false color composites. So I will assign this stack image into a new variable called d and run it. So now d is created d is showing here So let us first before exporting this stack amount let us try to plot it so is it proper or not, we can see here itself. So if you run the plot command the image will appear here. We can zoom it and see if you zoom here plotted image can be seen here. So if you want to see the specific bands, we can plot them also and you can see.

(Refer Slide Time: 22:37)



```
1 install.packages("raster")
2 install.packages("rgdal")
3
4 library(rgdal)
5 library(raster)
6
7 a <- raster(file.choose())
8 b <- raster(file.choose())
9 c <- raster(file.choose())
10 d <- stack(c,b,a)
11
12 plotRGB(d, r=1, g=2, b=3, stretch='lin')
13
14 writeRaster(d,filename = "C:/Users/Dr. Bharath H Aithal/Desktop/NPTEL/Datasets
15 |
```

The screenshot shows the RStudio interface. The console on the left contains the R code for installing and loading the 'raster' and 'rgdal' packages, and for reading three raster files into variables 'a', 'b', and 'c'. These are then stacked into a new raster 'd' using the 'stack' function. The 'd' raster is plotted using 'plotRGB' with a linear stretch, and the result is saved to a file using 'writeRaster'. The Environment pane on the right shows that variables 'a', 'b', 'c', and 'd' are all of the class 'RasterLayer'. The console output shows the execution of the code, including the file selection process for 'file.choose()' and the final file path for 'writeRaster'.

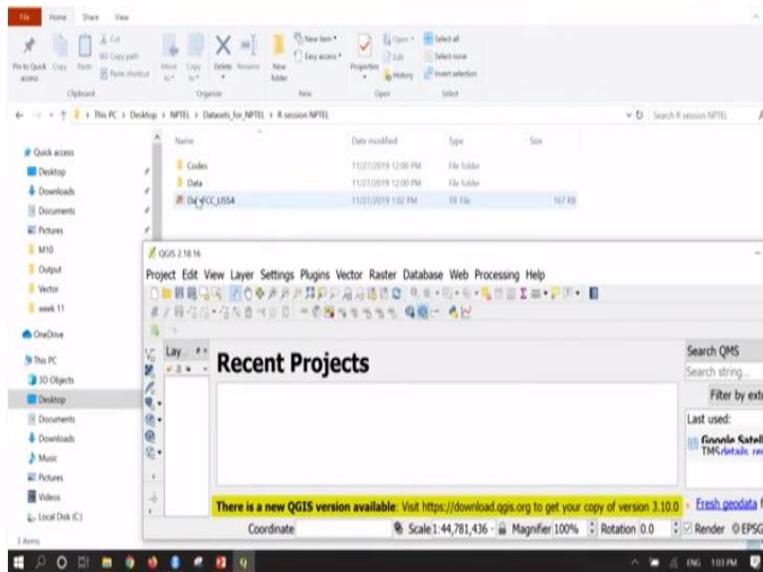
I will just plot band a and run it here we can see this is only band a so it can plot b this is band b and similarly band c also I can project if we collocate if you make a stack of these 3 things and make it a false color composite this will look something like this in which a vegetation looks in red color. To explore this particular d or the false color composite which we created we can use a command called writeRaster.

So there is a function called writeRaster so how exactly it works We can see go and check it in question mark writeRaster and run it. If you run that particular command write dot raster x comma file name, we need to mention. So here the file the x will get replaced by d and file name we are mentioning along with the path. So here since we have set the directory, we do not need to mention the path.

We just have to mention it as FCC and give the format geoTiff overwrite if an existing file is there, we will name it as FCC LISS 4 since this is a image from LISS4 imagery so I will rename it as LISS4 and then run. So now this particular LISS imagery has been exported into the folder. So where exactly it come up, we will try to see that. So, it must be in the folder import, it must be in the folder here if we set the directory so not here.

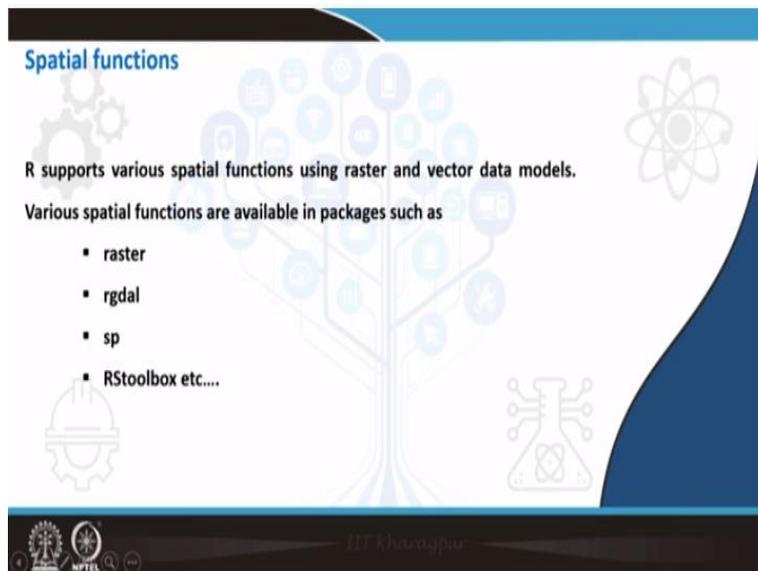
So since we need to mention the path here, let us mention the path and then run it again. So path is in R itself, we have the path. Let us go to the path; let us try to export it to this particular path to the data folder and then go to R file and then mention the path along with the filename. So now under the desktop NPTEL folder under data sets, this particular file will get saved. So I will just run it again. Now let us go and inspect the file is there or not?

(Refer Slide Time: 25:19)



It should be under data let us go and run it again. Okay R session NPTEL here it is the file has come up here. So since we are mentioned path till here. So let us go and see this particular false color composite in QGIS. In QGIS software we can just bring this particular file and see false color composite has been created.

(Refer Slide Time: 26:33)



In this particular session, we have seen how to install the packages how to run spatial functions.

(Refer Slide Time: 26:39)

Read raster data

- A function `raster()` is used from package "raster" to import satellite image from the directory to the R environment. Here `file.choose()` is a function can be used for choosing file instead of giving file path.
- Function `brick()` is a multilayer raster object, this function is used to call raster with multiple bands.

```
> quick_bird = raster(file.choose())
```

Or

```
> quick_bird = brick(file.choose())
```

And how to read raster data.

(Refer Slide Time: 26:40)

Cell statistics

To calculate minimum or maximum pixel value from a raster function `cellStats()` is used. This function also can be used to find mean and standard deviation from a raster object.

```
> min = cellStats(quick_bird, stat='min', na.rm=TRUE, asSample=TRUE)
```

```
> min  
[1] 123
```

```
> max = cellStats(image1, stat='max', na.rm=TRUE, asSample=TRUE)
```

```
> max  
[1] 1263
```

```
> mean = cellStats(image1, stat='mean', na.rm=TRUE, asSample=TRUE)
```

```
> mean  
[1] 432.3705
```

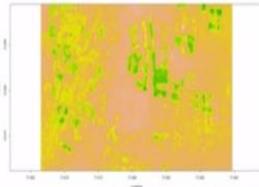
How to do cell statistics?

(Refer Slide Time: 26:41)

Plotting a raster map

Plotting a raster can be done using `plot()` function, adding of title, labels to x and y directions is possible.

```
> DEM = raster(file.choose())  
> plot(DEM, main = "Digital Elevation Model", xlab = "Longitude", ylab = "Latitude")
```



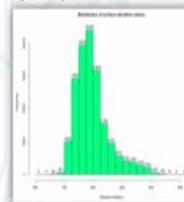
And plotting master data.

(Refer Slide Time: 26:44)

Preparation of histogram of a raster

Histogram of a raster file can be plotted with `hist()` function. Provision to customize the color, breaks and other parameters are incorporated in the function itself.

```
hist(DEM,main = "Distribution of surface elevation values",  
+ xlab = "Elevation (meters)",ylab = "Frequency",  
+ col = "springgreen",breaks=25,  
+ labels=TRUE,warn.unused=FALSE)
```



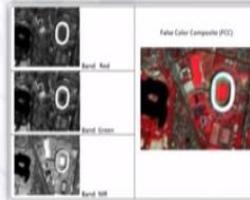
And creating histograms.

(Refer Slide Time: 26:47)

False Color Composite (FCC) preparation

Color composites from bands of satellite images can be formed by stacking the layers together in particular order and exporting or writing to various other formats is performed by using `writeRaster()` function.

```
a <- raster(file.choose())
> b <- raster(file.choose())
> c <- raster(file.choose())
> d <- stack(c,b,a)
> plotRGB(d, r=1, g=2, b=3, stretch='lin')
> writeRaster(d,"fcc.tif",format="GTiff",overwrite=TRUE)
```



And finally creating false color composite. So till here in the past 2 sessions we have seen installing packages, reading spatial data on the plotting maps and preparing false color composites or from different bands of the imagery being downloaded from the Bhuvan or any other geo portal. Thank you.