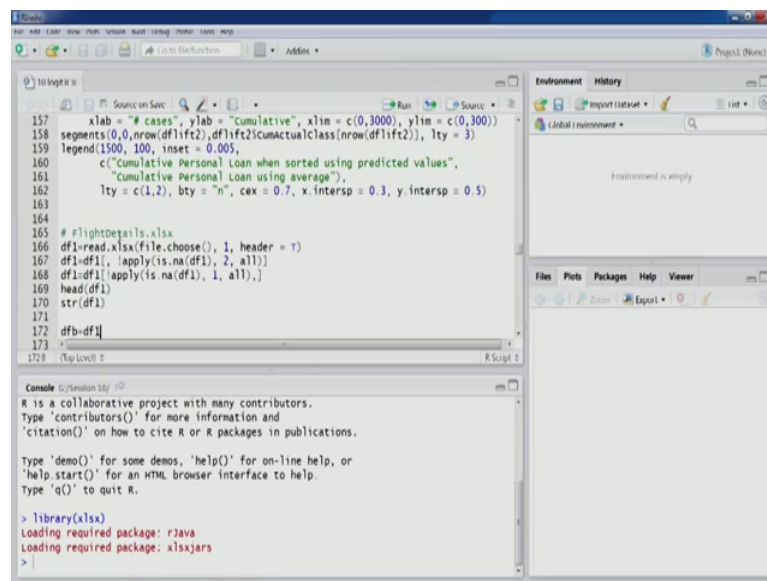


**Business Analytics & Data Mining Modeling Using R**  
**Dr. Gaurav Dixit**  
**Department of Management Studies**  
**Indian Institute of Technology, Roorkee**

**Lecture - 50**  
**Logistic Regression - Part V**

Welcome to welcome to the course Business Analytics and Data Mining Modeling Using R. So in previous few lectures we have been discussing Logistic Regression. So let us start our discussion from the same point where we left in previous lecture. So we were doing an exercise in R environment and we completed that exercise so that was using promotional offers; a data set. So what we are going to do in this particular lecture is we will use another data set and other problem and we will go through the complete analysis that is required in logistic regression.

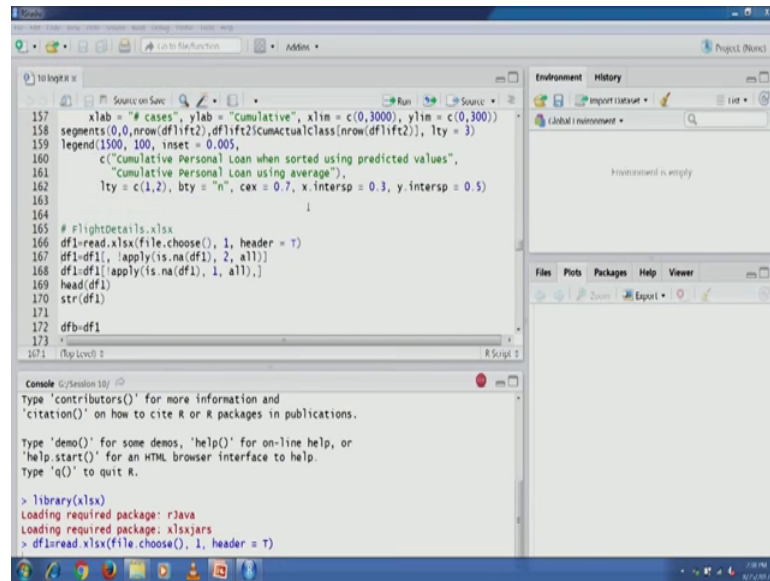
(Refer Slide Time: 00:49)



```
157 xlab = "# cases", ylab = "cumulative", xlim = c(0,3000), ylim = c(0,300))
158 segments(0,0,nrow(df1$ft2),df1$ft2$cumactualClass(nrow(df1$ft2)), lty = 3)
159 legend(1500, 100, inset = 0.005,
160       c("Cumulative Personal Loan when sorted using predicted values",
161         "Cumulative Personal Loan using average"),
162       lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # flightDetails.xlsx
166 df1<-read.xlsx(file.choose(), 1, header = T)
167 df1<-df1[, !apply(is.na(df1), 2, all)]
168 df1<-df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
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 the data set that we are going to use for this lectures exercise is on flight details. So this particular data set we have a used before as well. So let us so this is the data set; flight details dot xlsx. So, this particular data set have been used before. So let us load this, import this data set into R environment.

(Refer Slide Time: 01:26)



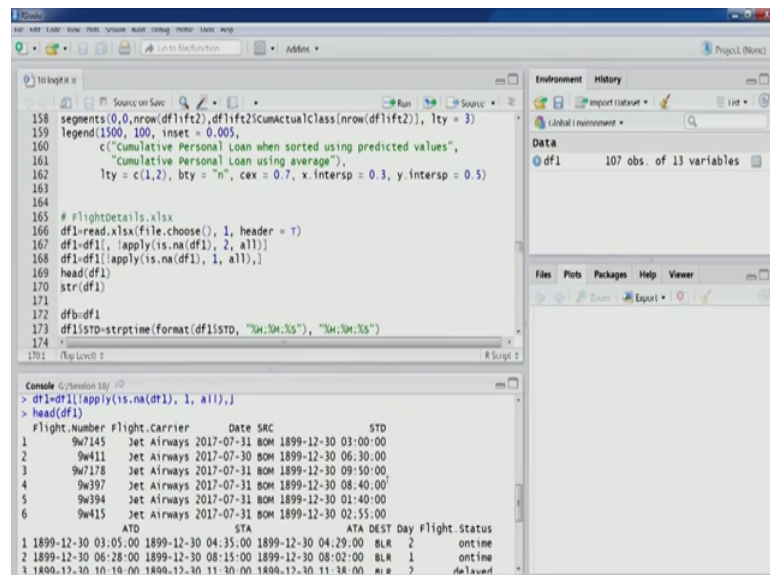
```
157 xlab = "# cases", ylab = "cumulative", xlim = c(0,3000), ylim = c(0,300))
158 segments(0,0,nrow(df1$ft2),df1$ft2$cumactualclass[nrow(df1$ft2)], lty = 3)
159 legend(1500, 100, inset = 0.005,
160       c("Cumulative Personal Loan when sorted using predicted values",
161         "Cumulative Personal Loan using average"),
162       lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # FlightDetails.xlsx
166 df1=read.xlsx(file.choose(), 1, header = T)
167 df1=df1[, !apply(is.na(df1), 2, all)]
168 df1=df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb=df1
173
174
```

Console

```
> library(xlsx)
Loading required package: rJava
Loading required package: xlsxjars
> df1=read.xlsx(file.choose(), 1, header = T)
```

So we have 108 observation 13 variables; however, there are some na rows and na columns. So let us get rid of them. So let us first remove na columns then na rows and you would see 107 observation 13 variables. Let us look at the first six observations. So this particular dataset we had used before when we discuss new base.

(Refer Slide Time: 01:50)



```
158 segments(0,0,nrow(df1$ft2),df1$ft2$cumactualclass[nrow(df1$ft2)], lty = 3)
159 legend(1500, 100, inset = 0.005,
160       c("Cumulative Personal Loan when sorted using predicted values",
161         "Cumulative Personal Loan using average"),
162       lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # FlightDetails.xlsx
166 df1=read.xlsx(file.choose(), 1, header = T)
167 df1=df1[, !apply(is.na(df1), 2, all)]
168 df1=df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb=df1
173 df1$STD=strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174
175
```

Console

```
> df1=df1[!apply(is.na(df1), 1, all),]
> head(df1)
  Flight.Number Flight.Carrier   Date SRC      STD
1      9w7145      Jet Airways 2017-07-31 BOM 1899-12-30 03:00:00
2      9w411      Jet Airways 2017-07-30 BOM 1899-12-30 06:30:00
3      9w7178      Jet Airways 2017-07-31 BOM 1899-12-30 09:50:00
4      9w397      Jet Airways 2017-07-31 BOM 1899-12-30 08:40:00
5      9w394      Jet Airways 2017-07-31 BOM 1899-12-30 01:40:00
6      9w415      Jet Airways 2017-07-31 BOM 1899-12-30 02:55:00
```

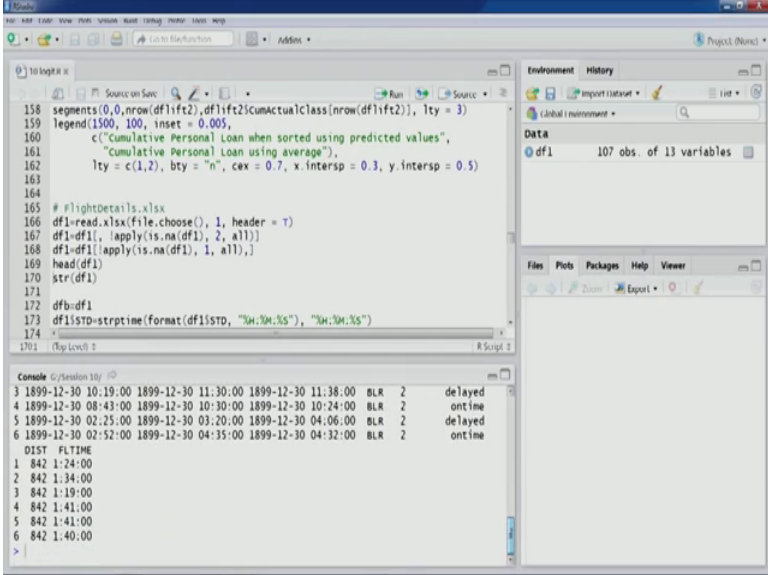
Data

```
df1 107 obs. of 13 variables
```

So now again we will understand; so this particular dataset most of the variables that we have they are factor variable, as you can see in first six observation. We have flight number, flight carrier, then date, then source and schedule time of departure, then we

have actual time of departure, then we have scheduled time of arrival, then actual time of arrival, then a destination, then we have day whether it was Sunday or Monday.

(Refer Slide Time: 02:16)



The screenshot shows the R Studio environment. The script editor contains R code for reading an Excel file, applying a function, and formatting dates. The console shows the output of the code, including a table of flight data with columns for date, time, flight number, and status.

```
158 segments(0,0,nrow(df1),df1$flight2,cumActualClass(nrow(df1), lty = 3))
159 legend(150, 100, inset = 0.005,
160       c("Cumulative Personal Loan when sorted using predicted values",
161         "Cumulative Personal Loan using average"),
162       lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # flightDetails.xlsx
166 df1<-read.xlsx(file.choose(), 1, header = T)
167 df1<-df1[, apply(is.na(df1), 2, all)]
168 df1<-df1[apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
173 df1$STO<-strptime(format(df1$STO, "%H:%M:%S"), "%H:%M:%S")
174
175
```

Console Output:

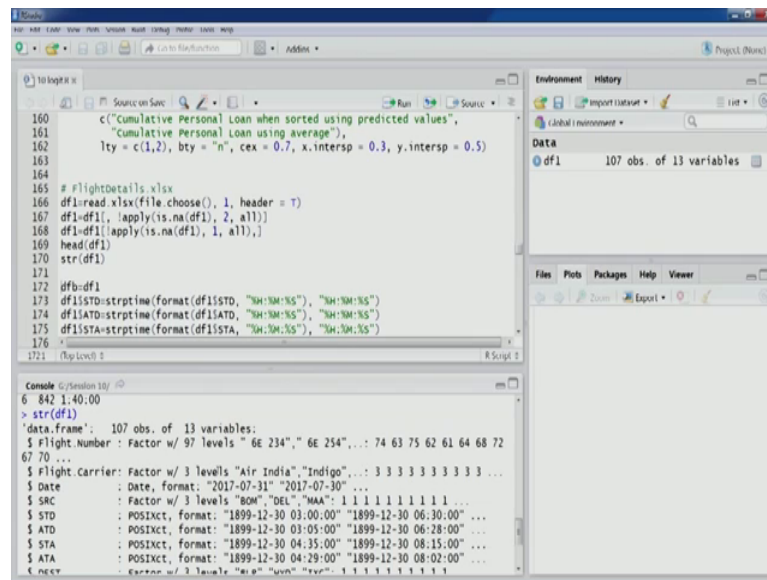
```
3 1899-12-30 10:19:00 1899-12-30 11:30:00 1899-12-30 11:38:00 BLR 2 delayed
4 1899-12-30 08:43:00 1899-12-30 10:30:00 1899-12-30 10:24:00 BLR 2 ontime
5 1899-12-30 02:25:00 1899-12-30 03:20:00 1899-12-30 04:06:00 BLR 2 delayed
6 1899-12-30 02:52:00 1899-12-30 04:35:00 1899-12-30 04:32:00 BLR 2 ontime
```

DIST	FLTIME
1	842 1:24:00
2	842 1:34:00
3	842 1:19:00
4	842 1:41:00
5	842 1:41:00
6	842 1:40:00

So we had information on just 2 days; data on just 2 days; then flight status and then two additional variables. So when we use this particular data set in a new base technique, we did not have these two variables; distance and flight time. So we have added these two variables distance between the source and destination and also flight time.

So this particular data set, so let us look at the structure and then we will discuss further. So let us look at this information. So, flight number if we see that the flight number is something that we are going not going to use in this particular analysis.

(Refer Slide Time: 03:07)

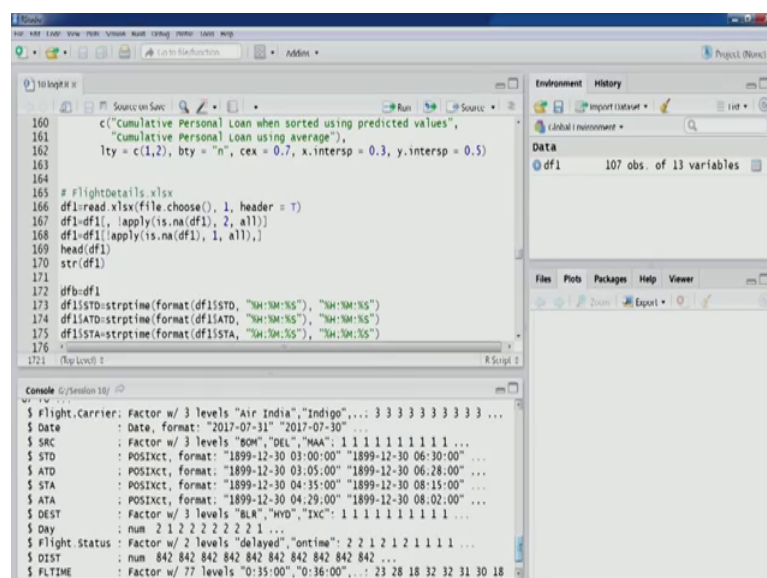


```
160 c("Cumulative Personal Loan when sorted using predicted values",
161   "Cumulative Personal Loan using average"),
162   lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # FlightDetails.xlsx
166 df1<-read.xlsx(file.choose(), 1, header = T)
167 df1<-df1[, !apply(is.na(df1), 2, all)]
168 df1<-df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
173 df1$STD<-strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174 df1$ATD<-strptime(format(df1$ATD, "%H:%M:%S"), "%H:%M:%S")
175 df1$STA<-strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
176
177
178 [Top Level] 2
```

```
6 842 1:40:00
> str(df1)
'data.frame': 107 obs. of 13 variables:
 $ Flight.Number: Factor w/ 97 levels "6E 234"," 6E 254",...: 74 63 75 62 61 64 68 72
 67 70 ...
 $ Flight.Carrier: Factor w/ 3 levels "Air India","Indigo",...: 3 3 3 3 3 3 3 3 ...
 $ Date: Date, format: "2017-07-31" "2017-07-30" ...
 $ SRC: Factor w/ 3 levels "BOM","DEL","MAA": 1 1 1 1 1 1 1 1 ...
 $ STD: POSIXct, format: "1899-12-30 03:00:00" "1899-12-30 06:30:00" ...
 $ ATD: POSIXct, format: "1899-12-30 03:05:00" "1899-12-30 06:28:00" ...
 $ STA: POSIXct, format: "1899-12-30 04:35:00" "1899-12-30 08:15:00" ...
 $ ATA: POSIXct, format: "1899-12-30 04:29:00" "1899-12-30 08:02:00" ...
 $ DEST: Factor w/ 3 levels "BLR","HYD","IXC": 1 1 1 1 1 1 1 1 ...
```

However, flight numbers for each of the you know flights that we have with us right; 97 of them out of 107 observations, some of them must be repeating. So then we have flight carrier so we have 3 carriers; Air India, Indigo and Jet Airways. So we will discuss them. Then we have date; so a data flight. We have flights on 2 days that is 30th July 2017 and 31st July 2017.

(Refer Slide Time: 03:35)



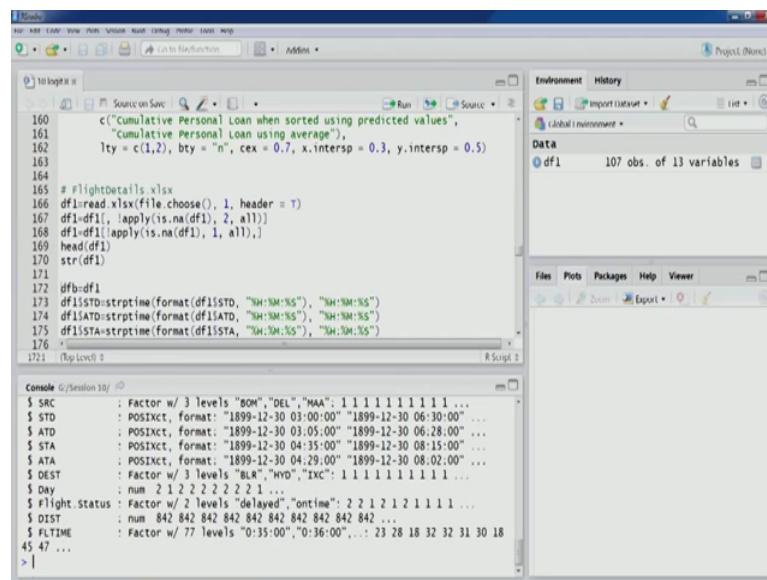
```
160 c("Cumulative Personal Loan when sorted using predicted values",
161   "Cumulative Personal Loan using average"),
162   lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # FlightDetails.xlsx
166 df1<-read.xlsx(file.choose(), 1, header = T)
167 df1<-df1[, !apply(is.na(df1), 2, all)]
168 df1<-df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
173 df1$STD<-strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174 df1$ATD<-strptime(format(df1$ATD, "%H:%M:%S"), "%H:%M:%S")
175 df1$STA<-strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
176
177
178 [Top Level] 2
```

```
$ Flight.Carrier: Factor w/ 3 levels "Air India","Indigo",...: 3 3 3 3 3 3 3 3 ...
 $ Date: Date, format: "2017-07-31" "2017-07-30" ...
 $ SRC: Factor w/ 3 levels "BOM","DEL","MAA": 1 1 1 1 1 1 1 1 ...
 $ STD: POSIXct, format: "1899-12-30 03:00:00" "1899-12-30 06:30:00" ...
 $ ATD: POSIXct, format: "1899-12-30 03:05:00" "1899-12-30 06:28:00" ...
 $ STA: POSIXct, format: "1899-12-30 04:35:00" "1899-12-30 08:15:00" ...
 $ ATA: POSIXct, format: "1899-12-30 04:29:00" "1899-12-30 08:02:00" ...
 $ DEST: Factor w/ 3 levels "BLR","HYD","IXC": 1 1 1 1 1 1 1 1 ...
 $ Day: num 2 1 2 2 2 2 2 1 ...
 $ Flight status: Factor w/ 2 levels "delayed","ontime": 2 2 1 2 1 2 1 1 ...
 $ DIST: num 842 842 842 842 842 842 842 ...
 $ FLTIME: Factor w/ 77 levels "0:35:00","0:36:00",...: 23 28 18 32 32 31 30 18
```

However, as we did in (Refer Time: 03:48) we would like we would not like to use this information and date is not important for our analysis and we will consider them you

know we will not considered date and we look at the time intervals of this specific time intervals, departure intervals of flights. So the main problem remains same. So this is a classification problem that we are going to model using logistic regression modeling. So where we would like to predict the delays of flights right.

(Refer Slide Time: 04:30)



```
160 c("Cumulative Personal Loan when sorted using predicted values",
161   "Cumulative Personal Loan using average"),
162   lty = c(1,2), bty = "n", cex = 0.7, x.intersp = 0.3, y.intersp = 0.5)
163
164
165 # FlightDetails.xlsx
166 df1=read.xlsx(file.choose(), 1, header = T)
167 df1=df1[, !apply(is.na(df1), 2, all)]
168 df1=df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dftb=df1
173 df1$STD=strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174 df1$ATD=strptime(format(df1$ATD, "%H:%M:%S"), "%H:%M:%S")
175 df1$STA=strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
176
177
178 (Top Level) >
```

Console

```
$ SRC      : Factor w/ 3 levels "BOM","DEL","MAA": 1 1 1 1 1 1 1 1 ...
$ STD      : POSIXct, format: "1899-12-30 03:00:00" "1899-12-30 06:30:00" ...
$ ATD      : POSIXct, format: "1899-12-30 03:05:00" "1899-12-30 06:28:00" ...
$ STA      : POSIXct, format: "1899-12-30 04:35:00" "1899-12-30 08:15:00" ...
$ ATA      : POSIXct, format: "1899-12-30 04:29:00" "1899-12-30 08:02:00" ...
$ DEST     : Factor w/ 3 levels "BLR","HYD","INC": 1 1 1 1 1 1 1 1 ...
$ Day      : num 2 1 2 2 2 2 2 2 1 ...
$ Flight status : Factor w/ 2 levels "delayed","ontime": 2 2 1 2 1 2 1 1 1 ...
$ DIST     : num 842 842 842 842 842 842 842 842 ...
$ FLTIME   : Factor w/ 77 levels "0:35:00","0:36:00",...: 23 28 18 32 32 31 30 18
45 47 ...
> |
```

So other variables we are familiar with a day and flight status, distance, flight time. So these are the other variables flight time. As you can see this is right now it is Factor, so this has to be converted into a numeric variable distances flight status is also ok. However, we would like to change the labels, you can see this is a delayed and ontime. However, since we are modeling for delayed, so we are trying to predict delays. So therefore, our reference category has to be ontime and modeling has to be done with respect to delayed. So we need to change these labels for our outcome variable that is flight status. So we would like to predict the flight status whether particular flight is going to be on time or delayed; focus is on delayed. So the ontime is going to be our reference category. Just like we used to have other techniques other examples, just like we used to have 1 and 0. So a focus was on always class 1; members belonging to class 1. So we always look to identify build a model which would be classify an observation into class 1.

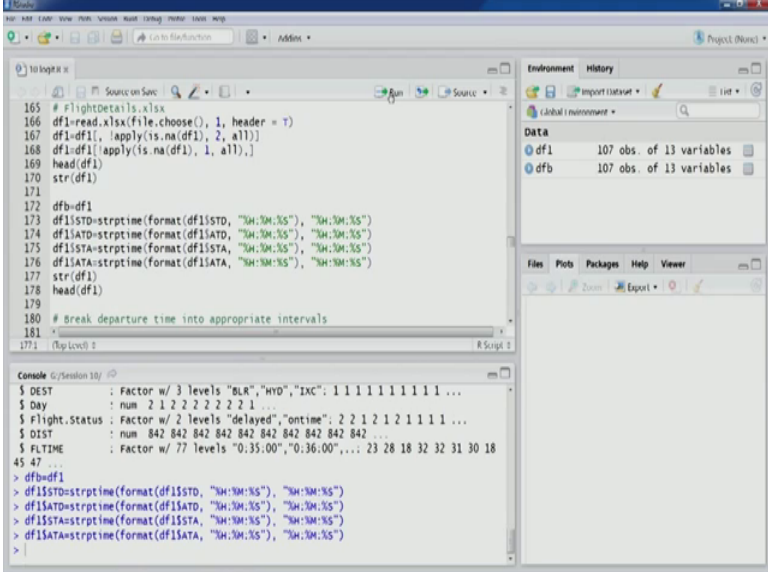
So, here in this case as I talked about we need to, we would we require to change the levels; destination is ok, day this also we would have to change. So day is factor variable,

2 days; Sunday and Monday. So right now this is numeric variable, so we will be required to change this variable as well. The schedule depart[ure]- you know departure time actual time of departure and these four information only will a try to derive another variable using these four information.

So flight status has been derived already been derived using some of these variables and we will derive another variable that departure time interval using some of these variables. Other after those derivation after those variable derivation we would not be using these variables.

Source is appropriately mentioned factor, 3 levels date we would not be using, flight carrier we would be using appropriately mention, flight number again we would not be using. So let us start some of these transformation. So before we go ahead, let us take a backup of this particular data set. So we will take a backup and then we would because as we said, we are not interested in actual dates of those flights.

(Refer Slide Time: 06:55)



```
165 # FlightDetails.xlsx
166 df1<-read.xlsx(file.choose(), 1, header = T)
167 df1<-df1[, !apply(is.na(df1), 2, all)]
168 df1<-df1[!apply(is.na(df1), 1, all),]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
173 df1$STD<-strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174 df1$ATO<-strptime(format(df1$ATO, "%H:%M:%S"), "%H:%M:%S")
175 df1$STA<-strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
176 df1$ATA<-strptime(format(df1$ATA, "%H:%M:%S"), "%H:%M:%S")
177 str(df1)
178 head(df1)
179
180 # Break departure time into appropriate intervals
181
1771 (Top Level) t
```

```
$ DEST      : Factor w/ 3 levels "BLR","HYD","INC": 1 1 1 1 1 1 1 1 1 ...
$ Day       : num  2 1 2 2 2 2 2 2 1 ...
$ Flight.Status: Factor w/ 2 levels "delayed","ontime": 2 2 1 2 1 2 1 1 1 ...
$ DIST      : num  842 842 842 842 842 842 842 842 ...
$ FLTIME    : Factor w/ 77 levels "0:35:00","0:36:00",...: 23 28 18 32 32 31 30 18
45 47
> dfb<-df1
> df1$STD<-strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
> df1$ATO<-strptime(format(df1$ATO, "%H:%M:%S"), "%H:%M:%S")
> df1$STA<-strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
> df1$ATA<-strptime(format(df1$ATA, "%H:%M:%S"), "%H:%M:%S")
>
```

So, I will like to change the same because we need to do, we need to use these particular columns for certain variable derivations. So we would like to change these dates to, so that it is it comes out to be the same date for all the flights and therefore, various derivation that we want to perform there are no issues in that.



(Refer Slide Time: 07:23)

```
166 df1<-read.xlsx(file.choose(), 1, header = T)
167 df1<-df1[, !apply(is.na(df1), 2, all)]
168 df1<-df1[, !apply(is.na(df1), 1, all)]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
173 df1$STD<-strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174 df1$ATD<-strptime(format(df1$ATD, "%H:%M:%S"), "%H:%M:%S")
175 df1$STA<-strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
176 df1$ATA<-strptime(format(df1$ATA, "%H:%M:%S"), "%H:%M:%S")
177 str(df1)
178 head(df1)
179
180 # Break departure time into appropriate intervals
181 range(df1$ATD)
182
183
```

Console Output:

```
$ SRC      : Factor w/ 3 levels "BOM","DEL","MAA": 1 1 1 1 1 1 1 1 1 ...
$ STD      : POSIXlt, format: "2017-08-25 03:00:00" "2017-08-25 06:30:00" ...
$ ATD      : POSIXlt, format: "2017-08-25 03:05:00" "2017-08-25 06:28:00" ...
$ STA      : POSIXlt, format: "2017-08-25 04:35:00" "2017-08-25 08:15:00" ...
$ ATA      : POSIXlt, format: "2017-08-25 04:29:00" "2017-08-25 08:02:00" ...
$ DEST     : Factor w/ 3 levels "BLR","HYD","INC": 1 1 1 1 1 1 1 1 1 ...
$ Day      : num 2 1 2 2 2 2 2 2 2 1 ...
$ Flight.status : Factor w/ 2 levels "delayed","ontime": 2 2 1 2 1 2 1 1 1 ...
$ DIST     : num 842 842 842 842 842 842 842 842 ...
$ FLTIME    : Factor w/ 77 levels "0:35:00","0:36:00",...: 23 28 18 32 32 31 30 18
45 47 ...
>
```

So, now you can see now in all these 4 dates, they have the dates has been changed. Earlier, it was earlier it was 1899; so this particular aspect we have already discussed during new base why 18 under 99 this particular date was coming there. Now this is what we require before we go for further variable transformation. So with this, so first six observation not much change only these four variables have been changed, dates have been change as you can see.

(Refer Slide Time: 08:00)

```
168 df1<-df1[, !apply(is.na(df1), 1, all)]
169 head(df1)
170 str(df1)
171
172 dfb<-df1
173 df1$STD<-strptime(format(df1$STD, "%H:%M:%S"), "%H:%M:%S")
174 df1$ATD<-strptime(format(df1$ATD, "%H:%M:%S"), "%H:%M:%S")
175 df1$STA<-strptime(format(df1$STA, "%H:%M:%S"), "%H:%M:%S")
176 df1$ATA<-strptime(format(df1$ATA, "%H:%M:%S"), "%H:%M:%S")
177 str(df1)
178 head(df1)
179
180 # Break departure time into appropriate intervals
181 range(df1$ATD)
182 breaks<-seq(strptime("00:00:00", "%H:%M:%S"), strptime("24:00:00", "%H:%M:%S"),
183             by = "6 hours")
184
```

Console Output:

```
5 9w118 Jet Airways 2017-07-31 BOM 2017-08-25 09:30:00
4 9w397 Jet Airways 2017-07-31 BOM 2017-08-25 08:40:00
5 9w394 Jet Airways 2017-07-31 BOM 2017-08-25 01:40:00
6 9w415 Jet Airways 2017-07-31 BOM 2017-08-25 02:55:00
ATD STA ATA DEST Day Flight.status
1 2017-08-25 03:05:00 2017-08-25 04:35:00 2017-08-25 04:29:00 BLR 2 ontime
2 2017-08-25 06:28:00 2017-08-25 08:15:00 2017-08-25 08:02:00 BLR 1 ontime
3 2017-08-25 10:19:00 2017-08-25 11:30:00 2017-08-25 11:38:00 BLR 2 delayed
4 2017-08-25 08:43:00 2017-08-25 10:30:00 2017-08-25 10:24:00 BLR 2 ontime
5 2017-08-25 02:25:00 2017-08-25 03:20:00 2017-08-25 04:06:00 BLR 2 delayed
6 2017-08-25 02:52:00 2017-08-25 04:35:00 2017-08-25 04:32:00 BLR 2 ontime
DIST FLTIME
1 842 1-24-00
```

Now, let us first variable transformation that we are going to perform is on departure time. So, we would like to break departure time in to appropriate intervals. So for example, let us look at the range now because now actual time of departure now. It is now you know we have excluded that date information now for all the observation we are using the same date. So therefore, range can be appropriately captured for our analysis.

(Refer Slide Time: 08:30)

```

175 df1$TA=strftime(format(df1$TA, "%H:%M:%S"), "%H:%M:%S")
176 df1$TA=strftime(format(df1$TA, "%H:%M:%S"), "%H:%M:%S")
177 str(df1)
178 head(df1)
179
180 # Break departure time into appropriate intervals
181 range(df1$TA)
182 breaks=seq(strptime("00:00:00", "%H:%M:%S"), strptime("24:00:00", "%H:%M:%S"),
183           by = "6 hours")
184 labels=c("0-6","6-12","12-18","18-24")
185 DEPT=cut(df1$TA, breaks = breaks, right = F, labels = labels)
186
187 df1=cbind(df1, DEPT)
188
189 df1$day=as.factor(df1$day)
190 levels(df1$day)
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

```

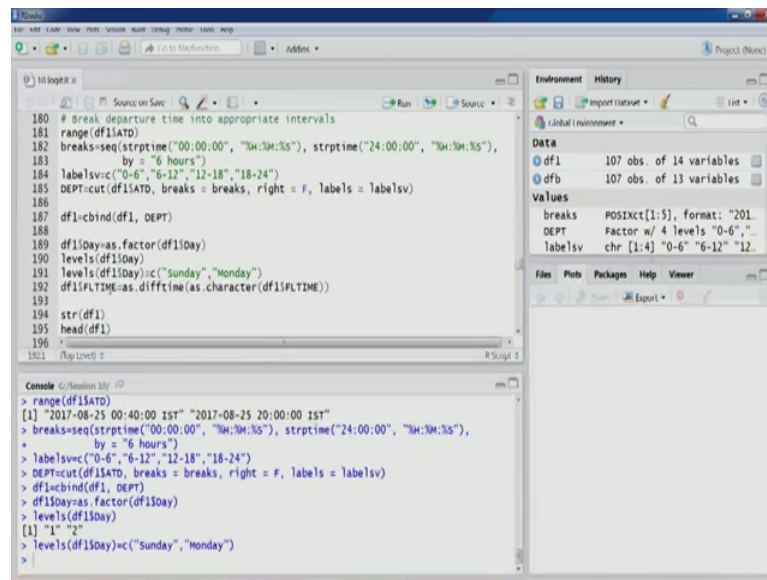
So, we have a flights ranging from flights ranging from flights ranging from 12:40 pm to 20 hours right. So, 40:00 hours 40 minutes to 20 hours right; so this is the range. Now we would like to break this time into appropriate intervals. So as we have done during new base we can see we would like to break this particular departure time into 4 intervals, you can see the labels also 0 to 6, 6 to 12, 12 to 18 and 18 to 24.

So, 24 hours time format; you would like to create four categories of that from 0 to 6 and 6 to 12 and 12 to 18, 18 to 12. So for this we would be requiring this particular breaks variable which would be used in the cut function as you can see in next line of code; this one that breaks is a breaks this particular variable is going to be used here and it will cut the different observation in actual time of departure using these breaks.

So let us compute this. So you would see in environment section, we will have breaks a variable and 5 values are there and let us also create labels.



(Refer Slide Time: 09:48)



The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains R code for creating time intervals and a data frame. The code includes comments and functions like `range`, `breaksseq`, `labelsvec`, `DEPTcut`, `df1cbind`, `df1Dayas.factor`, `levels`, and `df1FLTIMEas.difftime`.
- Environment:** Shows the global environment with variables `df1` (107 obs. of 14 variables) and `dfb` (107 obs. of 13 variables).
- Console:** Displays the output of the executed R code, showing the creation of the `df1` data frame and the resulting time intervals.

```
180 # Break departure time into appropriate intervals
181 range(df1$ATO)
182 breaks=seq(strptime("00:00:00", "%H:%M:%S"), strptime("24:00:00", "%H:%M:%S"),
183           by = "6 hours")
184 labelsvec=c("0-6","6-12","12-18","18-24")
185 DEPT=cut(df1$ATO, breaks = breaks, right = F, labels = labelsvec)
186
187 df1=cbind(df1, DEPT)
188 df1Day=as.factor(df1$Day)
189 levels(df1Day)
190 levels(df1Day)=c("Sunday", "Monday")
191 df1FLTIME=as.difftime(as.character(df1$FLTIME))
192
193 str(df1)
194 head(df1)
195
196
```

Console Output:

```
> range(df1$ATO)
[1] "2017-08-25 00:40:00 IST" "2017-08-25 20:00:00 IST"
> breaks=seq(strptime("00:00:00", "%H:%M:%S"), strptime("24:00:00", "%H:%M:%S"),
+           by = "6 hours")
> labelsvec=c("0-6","6-12","12-18","18-24")
> DEPT=cut(df1$ATO, breaks = breaks, right = F, labels = labelsvec)
> df1=cbind(df1, DEPT)
> df1Day=as.factor(df1$Day)
> levels(df1Day)
[1] "1" "2"
> levels(df1Day)=c("Sunday", "Monday")
>
```

So these breaks can be used to create these 4 categories. So, let us execute this. So you would see that depth this variable has been created with it is a factor with 4 levels right. So, we get it this particular variable into appropriate formats. Let us append this particular variable into the data frame. .

Now let us focus on other variables. So we had noticed that day was also you know day was we wanted to be a factor variable, categorical variable having a Sundays and Mondays flights on 2 day. However, this was stored as numeric. So let us change it to factor. Let us look at the level 1 and 2.

So we also would like to change these level names instead of 1 and 2 we would like to have Sunday and Monday; specifically specified. So we will do that. Once this is done we will focus on another variable that is flight time. So flight time that information we had in that time notation as hours colon minutes colon seconds. So, we would like to change it into a format that could be useful for our analysis. So, we would like to change them all those that this file flight duration flight time variable into minutes. So all those values we would like to convert into minutes.

So this is the function that can be used as dot difftime. So because these are you know time intervals, so their differences between two times. So this particular function as dot difftime can be used and you would see that we would be able to change it into minutes.

So first you would see that this particular variable has been changed to this. Let us look at this structure; so if you go to fl time you see that class difftime has been created.

(Refer Slide Time: 11:51)

The screenshot shows an R script with the following code:

```

183   by = "6 hours")
184   labels=c("0-6","6-12","12-18","18-24")
185   DEPT=cut(df1$ATD, breaks = breaks, right = F, labels = labels)
186
187   df1=cbind(df1, DEPT)
188
189   df1$Day=as.factor(df1$Day)
190   levels(df1$Day)
191   levels(df1$Day)=c("Sunday","Monday")
192   df1$FLTIME=as.difftime(as.character(df1$FLTIME))
193
194   str(df1)
195   head(df1)
196
197   dfb1=df1
198   df1=df1[,c(1,3,5:8)]
199
200

```

The console output shows the structure of the data frame:

```

$ ATD      : POSIXlt, format: "2017-08-25 03:05:00" "2017-08-25 06:28:00" ...
$ STA      : POSIXlt, format: "2017-08-25 04:35:00" "2017-08-25 08:15:00" ...
$ ATA      : POSIXlt, format: "2017-08-25 04:29:00" "2017-08-25 08:02:00" ...
$ DEST     : Factor w/ 3 levels "BLR","HYD","INC": 1 1 1 1 1 1 1 1 ...
$ Day      : Factor w/ 2 levels "Sunday","Monday": 2 1 2 2 2 2 2 2 1 ...
$ Flight.status : Factor w/ 2 levels "delayed","ontime": 2 1 2 1 2 1 1 1 ...
$ DIST     : num 842 842 842 842 842 842 842 ...
$ FLTIME   : class "difftime" atomic [1:107] 84 94 79 101 100 99 79 137 139 ...
..
.. attr(,"units")= chr "mins"
$ DEPT     : Factor w/ 4 levels "0-6","6-12","12-18",...: 1 2 2 2 1 1 2 2 2 ...

```

So this is now atomic factor and now all these values they are minutes right. So you can see character mins here this information, so all these values that were there time intervals. Now they have been converted into minutes.

(Refer Slide Time: 12:14)

The screenshot shows an R script with the following code:

```

185   DEPT=cut(df1$ATD, breaks = breaks, right = F, labels = labels)
186
187   df1=cbind(df1, DEPT)
188
189   df1$Day=as.factor(df1$Day)
190   levels(df1$Day)
191   levels(df1$Day)=c("Sunday","Monday")
192   df1$FLTIME=as.difftime(as.character(df1$FLTIME))
193
194   str(df1)
195   head(df1)
196
197   dfb1=df1
198   df1=df1[,c(1,3,5:8)]
199   str(df1)
200   head(df1)
201

```

The console output shows the structure of the data frame:

```

3 2017-08-25 10:19:00 2017-08-25 11:30:00 2017-08-25 11:38:00 BLR Monday
4 2017-08-25 08:43:00 2017-08-25 10:30:00 2017-08-25 10:24:00 BLR Monday
5 2017-08-25 02:25:00 2017-08-25 03:20:00 2017-08-25 04:06:00 BLR Monday
6 2017-08-25 02:52:00 2017-08-25 04:35:00 2017-08-25 04:32:00 BLR Monday
Flight.Status DIST FLTIME DEPT
1 ontime 842 84 mins 0-6
2 ontime 842 94 mins 6-12
3 delayed 842 79 mins 6-12
4 ontime 842 101 mins 6-12
5 delayed 842 101 mins 0-6
6 ontime 842 100 mins 0-6

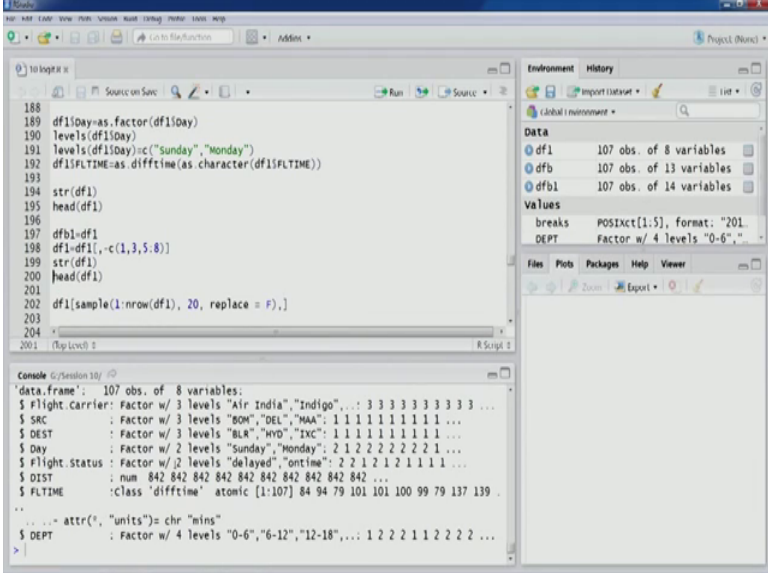
```

So with this almost we are almost there. So you can see that flight time; now you can see 84 minutes, 94 minutes, 79 minutes. So all the flight time duration have been

appropriately converted. We have created the departure variable, each of the flight has been correctly labeled as per it is newly created category in departure and so let us focus on some other transformations; so before that a let us take a backup. .

Now as we have talked about that some of these variables we would not be considering. So for example, variable 1 that is flight number, then date that is 3 column number 3; we would not consider, then from column number 5 to 8. So these are the actual dates scheduled time of departure, actual time of departure, scheduled time of arrival and actual time of arrival. So these also, these 4 columns also we would not be taking into model.

(Refer Slide Time: 13:17)



```
188 df1$day=as.factor(df1$day)
189 levels(df1$day)
190 levels(df1$day)=c("Sunday","Monday")
191 df1$FLTIME=as.difftime(as.character(df1$FLTIME))
192
193
194 str(df1)
195 head(df1)
196
197 df1=df1[-c(1,3,5,8)]
198 df1=df1[sample(1:nrow(df1), 20, replace = F),]
199 str(df1)
200 head(df1)
201
202
203
204
```

Environment History

Data

- df1 107 obs. of 8 variables
- dfb 107 obs. of 13 variables
- dfbl 107 obs. of 14 variables

Values

- breaks POSIXct[1:5], format: "201"
- DEPT Factor w/ 4 levels "0-6",...

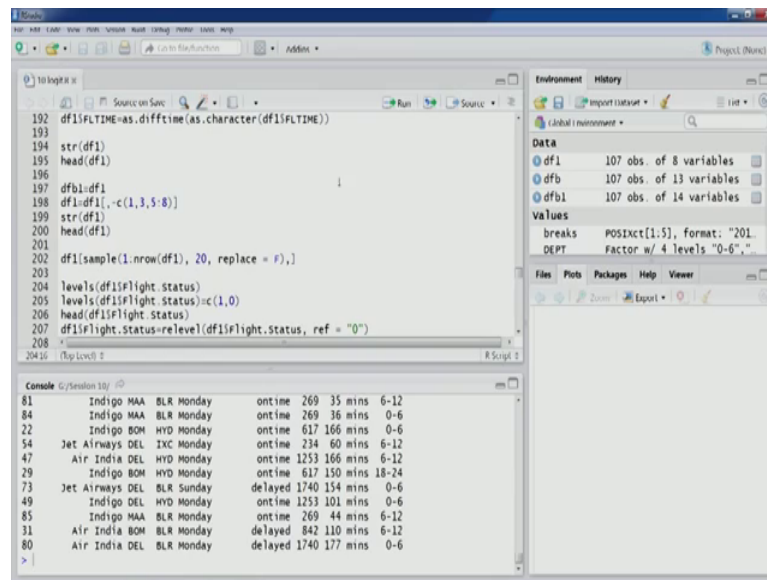
Console

```
data.frame': 107 obs. of 8 variables:
 $ flight.carrier: Factor w/ 3 levels "Air India","Indigo",...: 3 3 3 3 3 3 3 3 3 ...
 $ SRC          : Factor w/ 3 levels "BOM","DEL","MAA": 1 1 1 1 1 1 1 1 1 ...
 $ DEST        : Factor w/ 3 levels "BLR","HYD","INC": 1 1 1 1 1 1 1 1 1 ...
 $ Day         : Factor w/ 2 levels "Sunday","Monday": 2 1 2 2 2 2 2 2 1 ...
 $ Flight.status: Factor w/ 2 levels "delayed","ontime": 2 2 1 2 1 2 1 1 1 ...
 $ DIST        : num 842 842 842 842 842 842 842 842 ...
 $ FLTIME      : class 'difftime' atomic [1:107] 84 94 79 101 100 99 79 137 139 ...
 .. attr(,"units")= chr "mins"
 $ DEPT        : Factor w/ 4 levels "0-6","6-12","12-18",...: 1 2 2 2 1 1 2 2 2 ...
```

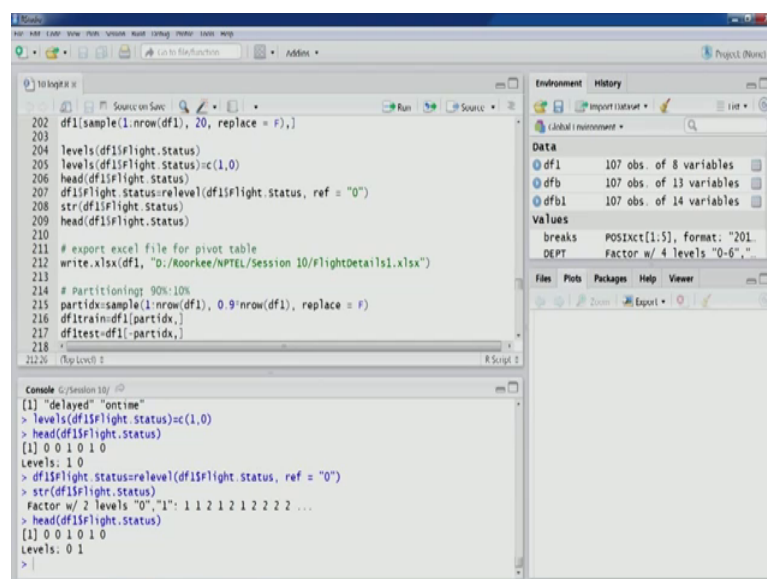
So let us get rid of these variables, these columns. Now so these are the variables that we are left with and also they are in appropriate format. So we would like to use these variables in our logistic regression model. So we now we have flight carrier, source, destination, day, flight status, our outcome variable; however, we need to change the levels as we talked about; distance and we have flight time, minutes and then we have this departure interval.

So once this is done and these are first six observations. If we want to take you know any random sample of 20; so this is sample function can be used in this fashion. So this particular command will give us 20 randomly selected rows. So we can have a look at different values for different observations randomly drawn observations.

(Refer Slide Time: 14:12)



(Refer Slide Time: 14:19)



Now, let us focus on the outcome variable. So outcome variable, the levels are delayed and ontime in that particular order. Now we would like to change this into numeric code because later on the different later on our code would be a much easier to write, if we have the numeric codes right because we would be creating lift curve and cumulative lift curve we would be we would require numeric code, so that we can create the cumulative actual class. So all those things for all those things we would prefer this level for outcome variable to be 1 and 0.

So let us change this. So 1, here as you can see; 1 is corresponding to delayed and 0 is corresponding to ontime. Because our task is to predict delayed flights right. So therefore, one has to be assigned to delayed right. So let us execute this. Let us look at the first six observation of this outcome variable, flight status; you can see levels have changed now 1 to 0. However, if you see that ordering is 1 and 0. So 1 would become reference category and that means the rate would become reference category. So we did not want that. So we would like to change this. So re level function can be used to perform this kind of change you can see. In the re level first you need you need to pass on the factor variable and then the second argument is for the reference. So, we can select the reference category here, the other things would be appropriately changed.

So, I be execute this code and look at the structure of this particular variable. Now you can see 0 and 1; in the correct order and 1 representing now delayed and 0 representing ontime flights. So now, this particular variable; let us look at first six observations also. Now this particular variable is in the desired you know state as we want it to be. Now at this moment if we want certain analysis; descriptive analysis, we can do so as we did in new base as well. We can once this data is ready for logistic regression model, all the key variables are there and the data frame; final data frame. Then we can write this particular file into our disk and then we can apply you know pivot table excel, base pivot tables to write some summary to for further analysis.

So let us, so I have already created one pivot table. So let us open this. So this was the data; pivot table this particular exercise we have already done.

(Refer Slide Time: 16:58)

Flight Carrier	SRC	DEST	Day	Flight Status	DIST	FLTIME	DEPT
Jet Airways	DOM	DLR	Monday	0	812	81	0-6
Jet Airways	DOM	DLR	Sunday	0	812	91	6-12
Jet Airways	DOM	DLR	Monday	1	812	79	6-12
Jet Airways	DOM	DLR	Monday	0	812	101	6-12
Jet Airways	DOM	DLR	Monday	1	812	101	0-6
Jet Airways	DOM	DLR	Monday	0	812	100	0-6
Jet Airways	DOM	DLR	Monday	1	812	99	6-12
Jet Airways	DOM	DLR	Monday	1	812	79	6-12
Jet Airways	DOM	DLR	Monday	1	812	137	6-12
Jet Airways	DOM	DLR	Sunday	1	812	139	6-12
Jet Airways	DOM	IND	Monday	1	1251	110	6-12
Jet Airways	DOM	IND	Monday	0	1251	121	0-6
Jet Airways	DOM	IND	Monday	0	617	81	0-6
Air India	DOM	IND	Monday	0	617	89	6-12
Jet Airways	DOM	IND	Monday	0	617	119	0-6
Air India	DOM	IND	Monday	1	617	89	6-12
Jet Airways	DOM	IND	Monday	1	617	80	6-12
Jet Airways	DOM	IND	Monday	1	617	90	6-12
Jet Airways	DOM	IND	Sunday	1	617	81	0-6
Air India	DOM	IND	Sunday	1	617	86	0-6
Jet Airways	DOM	IND	Monday	1	617	171	0-6
Indigo	DOM	IND	Monday	0	617	166	0-6
Jet Airways	DOM	IND	Monday	0	617	191	0-6
Air India	DOM	IND	Monday	0	617	116	6-12
Jet Airways	DOM	IND	Monday	1	617	188	18-21
Air India	DOM	IND	Monday	1	617	212	18-21
Jet Airways	DOM	IND	Monday	0	617	106	0-6
Indigo	DOM	IND	Monday	0	617	170	12-18
Indigo	DOM	IND	Monday	0	617	150	18-21
Indigo	DOM	IND	Monday	1	617	176	18-21
Air India	DOM	DLR	Monday	1	812	110	6-12
Air India	DOM	DLR	Sunday	1	812	11	6-12
Air India	DOM	DLR	Monday	1	812	131	0-6

So we have to just select the data and you know get it into appropriate presentable format and then select all the all the columns, all the rows and then create pivot table using the insert tab and within that this pivot table option and the pivot table would be ready for us in this fashion.

(Refer Slide Time: 17:11)

SRC	Jet Airways	Air India	Indigo	Grand Total
DOM	5	2	9	16
DLR	5	3	7	15
IND	2	5	7	14
Grand Total	12	10	23	45

So, I have already selected few of the important variables here, in appropriate filters. So you can see in row levels I have, in row level I have SRC that is source; then in column level I have flight carrier those three flight carrier that we have; Air India, Indigo and Jet



Airways and three source we have a that is BOM, DEL and MAA that is Mumbai, Delhi, Mumbai airport, Delhi airport and Madras airport. And then we have in the values areas we have account of flight status that means, how many flights are actually delayed.

So how many flights are there; count of flight. So overall, now however, in the report filter I have flight status and day as you can see here. So these are report filters point. So flight status you can see I have pre selected one. So this can be changed if we click on this filter you can see all the flights count of all the flights or delayed flight 1 representing delayed. So therefore delayed flights and 0 representing on time so that can be done. .

In day also we have a filter. So this is since this is report filter, so we are going to have a filter for all the variables. So Sunday and Monday you can see here that we can have all you know all days Sundays and Monday's together total and then either Sunday or Monday. So those numbers would be reflected. So we these this particular descriptive table will change depending on the change in levels of report filter variable.

So let us look at some of these numbers. So as we can see that when we are interested in finding the count of delayed flights, you can see the flight status is selected as 1 right here; in the filter flight status is selected as 1. Therefore, the we are interested in understanding the delayed flights from these descriptive statistics, summery statistics. So we can see that then off further we have selected Monday. So delayed flights and then Monday; so these are the numbers. So we can see when the source is Mumbai airport that is BOM, then we can see total number of delayed flights are 16 and you would see Jet Airways more number of flights of Jet ways Jet Airways are delayed.

(Refer Slide Time: 19:56)

Flight Carrier	Src	Dst	Day	Flights Status	Dst	Estimated Time
Jet Airways	DOM	HYD	Monday	1	617	188 18-21
Jet Airways	DOM	HYD	Monday	1	617	90 6-12
Jet Airways	DOM	DLR	Monday	1	812	79 6-12
Jet Airways	DOM	HYD	Monday	1	617	80 6-12
Jet Airways	DOM	DLR	Monday	1	812	101 0-6
Jet Airways	DOM	DEL	Monday	1	1211	140 6-12
Jet Airways	DOM	DLR	Monday	1	812	90 6-12
Jet Airways	DOM	DLR	Monday	1	812	79 6-12
Jet Airways	DOM	DLR	Monday	1	812	137 6-12

So in this table once you click you will get the specific observations as well right. So now if we look at, if we look at the second row we see that Delhi. So overall total number of delayed flights for Mumbai and airport both these airports. So these airports are too busy airports in our country and you can see number of delayed air fights are similar number is there. And Madras airport we have just 7 delayed flights; however, across 3 carriers we see that Jet Airways we have more number of flights; in terms of number we have known more number of delayed flights.

So, in terms of just the number right; in terms of just the number and this information is for Monday. So now what we can do is a because with respect to delayed flights, let us look at the what happens during Sundays. Let us select Sunday and we do; so now you would see during Sundays we did not have any flight in our data set originating from Madras airport; so that is gone.

(Refer Slide Time: 21:04)

Flight Status	Day	SRC	Air India	Indigo	Jet Airways	Grand Total
1	Sunday		2	1	2	5
2					2	2
3			2	1	4	7

So that is also one problem in modeling exercise as I have I have talked about in previous lectures as well; that if some of the combinations are not covered then that could be a problem when we go about predicting new observation because if that observation falls into that zone and in our training partition that was not covered; some of those combinations were not covered in our training partitions then prediction would not be possible.

So the same thing is reflected here. For example, Sunday we do not have resource destination MMA Madras. We do not have any flights from that source and if we look at the number of flights are quite few in comparison to Monday. However, if we look at again more number of delayed flights or from Mumbai and again more number of delayed flights are from Jet Airways. So, in terms of numbers total number of flights are also less and during Sundays and total number of therefore total number of delayed flights are also less.

So, if we look proportion wise then the that does not seem to be much difference between Sunday and Monday. So the same thing we should expect when we build our logistic regression model, the same thing we should be expecting in our results as well; that there in terms of you know we look proportion wise.

Then there is seem to be any difference in delayed flights, you know whether it is Sunday or Monday. The similar kind of exercise can also be performed for ontime

flights. So we can go to flight status report filter, we can select 0 and that would represent the ontime flights. Now we can see the Sunday numbers here and Bombay 3 flights ontime, Delhi 2 flights ontime and the Madras 1 flight ontime. So we did have a flight on Sunday from Madras, but that was ontime.

So we can change the day filter as well to a Monday to see the difference. So we expect more number of flights during Monday. So that is the case. So we see that a more number of flights; 15, 22, 19. However, if we really look at these numbers during Mondays we see that there are more number of flights, there are more number of flights from Indigo which seem to be on time. However proportional wise our number of flights that they are running on Mondays are also more; are also higher, so that could be the region.

So this kind of analysis we can do using pivot tables and that would give us some insights when we go into former technique like logistic regression, what we should be expecting. So this kind of analysis can also help us in grouping as we have been talking about in previous lecture; grouping some of the categories on this we would be able to understand that which source destination can be grouped or which days can be grouped here, we just have 2 days. So therefore, the question would be whether we should incorporate day at all? If it is, the if does not seem to be a significant predictor for delayed flights as per the descriptive stats that we saw.

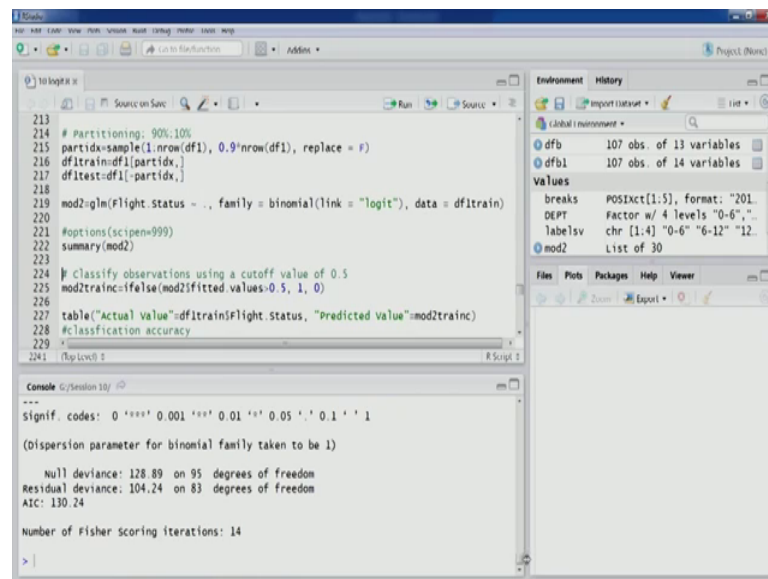
So these kind of decisions can be these kind of insights can be derived from descriptive stats. So what we will do? We will go back to R environment and we will move to our next step that is a partitioning. So because this particular data set is quite small; just 107 observations and for training partition you would like to have more number of observations; so that the model is slightly a more stable, more number of combinations because we are using a factor in many.

There are you know majority of the predictors that have they are factor they are categorical. So there you know there are going to be more combinations of values that you would like to cover in our model. So more observation we would like to have in the training partition because of this small sample size.

So let us do this partition partitioning; 90 percent for training partition and the remaining 10 percent for testing. You can see our partitions are created; 96 observation in the

training partition and 11 observation, remaining 11 observations in test partition. Now as we did in previous lecture, the same function glm can be used. Flight status is our outcome variable, other variables are going to be going to be used as predictors other things remain same. So let us run this code. You get the model. Let us look at the summary.

(Refer Slide Time: 26:11)



```

213 # Partitioning: 90%:10%
214 partidx=sample(1:nrow(df1), 0.9*nrow(df1), replace = F)
215 dfitrain=df1[partidx,]
216 dfitest=df1[-partidx,]
217
218 mod2=glm(Flight.Status ~ ., family = binomial(link = "logit"), data = dfitrain)
219
220 #options(scipen=999)
221 summary(mod2)
222
223 # Classify observations using a cutoff value of 0.5
224 mod2trainc=ifelse(mod2$fitted.values>0.5, 1, 0)
225
226 table("Actual Value"=dfitrain$Flight.status, "Predicted Value"=mod2trainc)
227 #classification accuracy
228
229
230
231
232
233
234
235
236
237
238
239
240
241

```

```

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

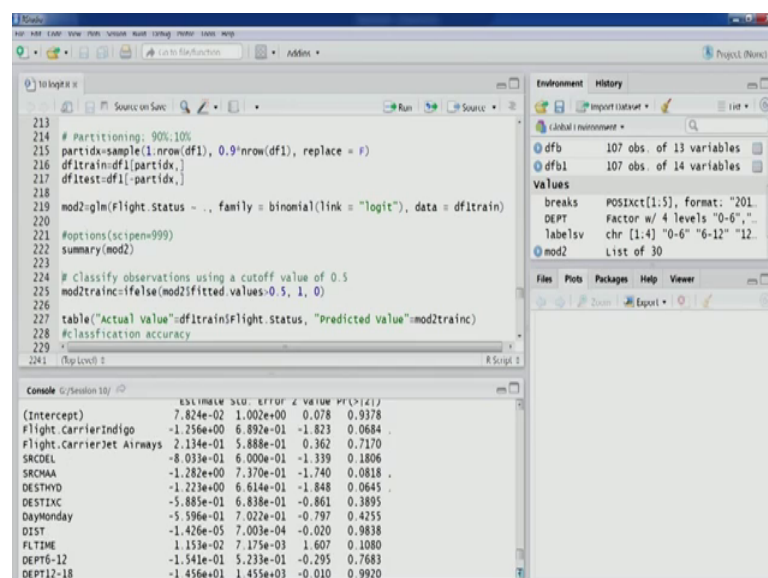
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 128.89  on 95  degrees of freedom
Residual deviance: 104.24  on 83  degrees of freedom
AIC: 130.24

Number of Fisher Scoring iterations: 14

```

(Refer Slide Time: 26:16)



```

213 # Partitioning: 90%:10%
214 partidx=sample(1:nrow(df1), 0.9*nrow(df1), replace = F)
215 dfitrain=df1[partidx,]
216 dfitest=df1[-partidx,]
217
218 mod2=glm(Flight.Status ~ ., family = binomial(link = "logit"), data = dfitrain)
219
220 #options(scipen=999)
221 summary(mod2)
222
223 # Classify observations using a cutoff value of 0.5
224 mod2trainc=ifelse(mod2$fitted.values>0.5, 1, 0)
225
226 table("Actual Value"=dfitrain$Flight.status, "Predicted Value"=mod2trainc)
227 #classification accuracy
228
229
230
231
232
233
234
235
236
237
238
239
240
241

```

```

              (Intercept)      Flight.carrierIndigo      Flight.carrierJet Airways      SRCDL      SRCHMA      DESTHYD      DESTXC      DayMonday      DIST      FLTIME      DEPT6-12      DEPT12-18
              7.824e-02      -1.256e+00      2.134e-01      -8.033e-01      -1.282e+00      -1.223e+00      -5.885e-01      -5.596e-01      -1.426e-05      1.153e-02      -1.541e-01      -1.456e+01
              1.002e+00      6.892e-01      5.888e-01      6.000e-01      7.370e-01      6.614e-01      6.838e-01      7.022e-01      7.003e-04      7.175e-03      5.233e-01      1.455e+03
              0.078      -1.823      0.362      -1.339      -1.740      -1.848      -0.861      -0.797      -0.020      1.607      -0.295      -0.010
              0.9378      0.0684      0.7170      0.1806      0.0818      0.0645      0.3895      0.4255      0.9838      0.1080      0.7683      0.9920

```

So if we look at the results of this logistic regression model that we have just created. We can see the results most of the variables because of this smaller data size, we do not

see much significance. However, we do see three variables to be significant at 90 percent confidence interval. You can see small dot here that is for 90 percent confidence interval. So, these three variables seem to be significant; first one is Flight Carrier Indigo.

So, the so the estimate is negative. So this seems to be significant at 90 percent confidence interval and so therefore what we can understand is with reference to Indigo, with reference to Air India which is the reference category for flight carrier; Indigo of flights from Indigo flights from indigo carrier. They seem to be less delayed because remember this particular modeling exercise with respect to delayed flights.

So therefore, if the coefficient is negative, so therefore we should expect that logit values are going to be that logit value there is going to be less cost, there is going to be decrease and therefore there is decrease in probabilities value and therefore there is more chance for it to be ontime rather than delayed.

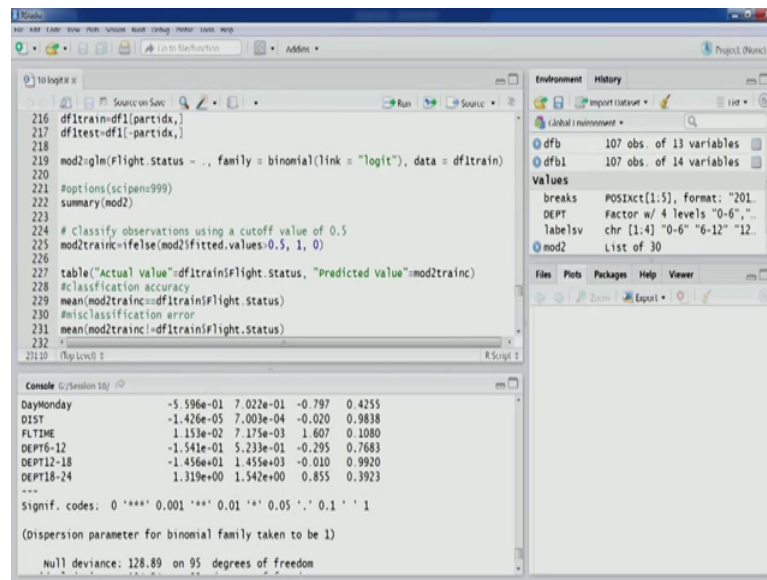
So, in this fashion we can interpret. However, the append be interpret these (Refer Time: 27:46) We should also look at first whether the variable is significant. So we have to check the signal as per our accepted level of confidence interval. We can look at the significance and once the variables are a significant then we can look at their coefficient values to understand the level of impact that they have. .

So flight from Indigo they seem to be less delayed or more on time in comparison to Air India right. The same cannot be said about Jet Airways because this is insignificant relationship. Now, other very other significant coefficient is source that is from Madras airport. So it seems that flights which originate from a Madras airport they also seem to be you know a less delayed right so more on time.

So this is also again at 90 percent confidence interval. Similarly we have another significant variable that is destination Hyderabad. So this is dummy code for Hyderabad. So airport, so we can see here also that the flights which arrive at Hyderabad; so where the destination is Hyderabad. They seem to be less delayed or more ontime with respect to our reference category. Same is also that Madras airport was also with respect to the reference category.



(Refer Slide Time: 29:19)



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

```
216 dfitrain=df1[partidx,]
217 dfitest=df1[-partidx,]
218
219 mod2=glm(Flight.Status ~ ., family = binomial(link = "logit"), data = dfitrain)
220
221 #options(scipens=999)
222 summary(mod2)
223
224 # Classify observations using a cutoff value of 0.5
225 mod2train=ifelse(mod2$fitted.values>0.5, 1, 0)
226
227 table("Actual Value"=dfitrain$Flight.Status, "Predicted Value"=mod2train)
228 #classification accuracy
229 mean(mod2train==dfitrain$Flight.Status)
230 #misclassification error
231 mean(mod2train!=dfitrain$Flight.Status)
232
```

The console output shows the summary of the logistic regression model:

```
Console 6/Session 10/ /R
DayMonday      -5.596e-01  7.022e-01  -0.797  0.4255
DIST           -1.426e-05  7.003e-04  -0.020  0.9838
FLTIME          1.153e-02  7.175e-03   1.607  0.1080
DEPT6-12        -1.541e-01  5.233e-01  -0.295  0.7683
DEPT12-18       -1.456e+01  1.455e+03  -0.010  0.9920
DEPT18-24       1.319e+00  1.542e+00   0.855  0.3923
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 128.89  on 95  degrees of freedom
```

The Environment pane on the right shows the following objects:

- dfb: 107 obs. of 13 variables
- dfb1: 107 obs. of 14 variables
- mod2: List of 30

Now other variables we can see they do not seem to be same even. For example, day Monday; so as we saw in our pivot table that a proportion wise there did not seem to be much difference on in on flights during Mondays or Sundays. So this also, in the results also logistic regression model results also this particular variable dummy variable does not come out to be significant.

So we can see that what we expected from our pivot table exercise, is the same thing is reflected here in the model. A distance this also is highly insignificant; so distance is not a the key predictor here. So this can also we so from this we can also understand which variable can be dropped and another modeling could be done. For example, distance is highly insignificant. So probably this variable can be dropped.

However, as I have pointed out in earlier lectures also, in data mining modeling our goal is prediction. So we are not even if a particular predictor is insignificant, but it is of practical importance in terms of predicting tasks, you know prediction or classification tasks or other data mining tasks; we would still like to keep it in the model.

However, in this case distance does not seem to be of much practical importance and highly insignificant. So probably this can be dropped. Flight time we can see that it was you know quite close to being significant at 90 percent confidence interval. So probably a flight time is also quite practical in terms of predicting delayed flights. So this we should anyway keep there and the then this next one is departure time interval.

So, with respect to the reference category that is 0 to 6 and these 3 departure interval they do not seem to be significant.

So probably at the departure time intervals also do not matter with respect to the data set that we have. So what we can do? We can look at the another modeling approach. So from this model we can further understand which variables are important; which variable are insignificant and if they are significant what is the impact that they have and whether another model with only the important variables can be done. Even if a particular variable is insignificant as we talked about it can still be kept in the model, if it is it provides some practical importance. So with this we will stop here and we will continue our discussion on this particular modeling exercise on flight delays.

Thank you.