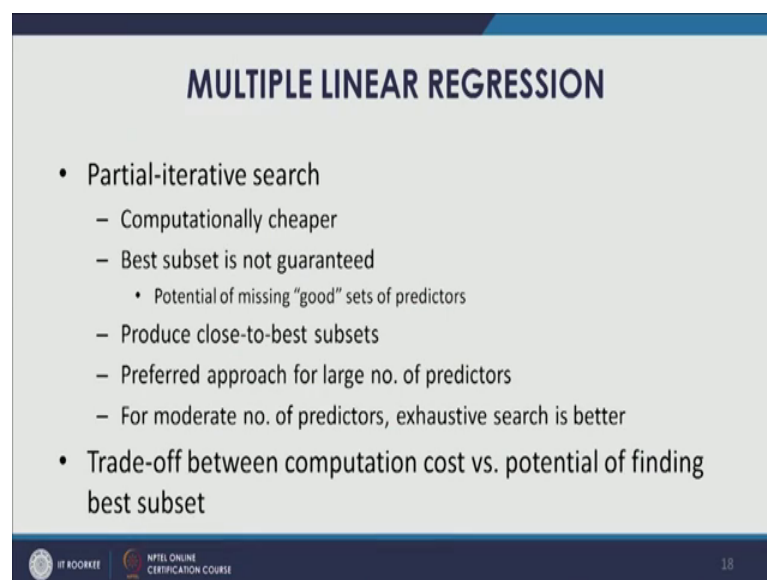


**Business Analytics & Data Mining Modeling Using R**  
**Dr. Gaurav Dixit**  
**Department of Management Studies**  
**Indian Institute of Technology, Roorkee**

**Lecture – 27**  
**Multiple Linear Regression–Part VI Partial Iterative Search**

Welcome to the course business analytics and data mining modeling using R. So, in the previous lecture, we were discussing a multiple linear regression. And we concluded our discussion on exhaustive search. So, the next approach that we use for variable selection and also for dimension reduction is the partial iterative search. So, there are a few algorithms under this particular approach that we are going to cover.

(Refer Slide Time: 00:43)



**MULTIPLE LINEAR REGRESSION**

- Partial-iterative search
  - Computationally cheaper
  - Best subset is not guaranteed
    - Potential of missing “good” sets of predictors
  - Produce close-to-best subsets
  - Preferred approach for large no. of predictors
  - For moderate no. of predictors, exhaustive search is better
- Trade-off between computation cost vs. potential of finding best subset

IT ROORKEE    NPTEL ONLINE CERTIFICATION COURSE    18

So, let us start our discussion. So, partial iterative search; this is a computationally cheaper in comparison to the exhaustive search that we do. So, in exhaustive search we try out all possible combinations of predictors. So, therefore, it is more like a brute force approach, and therefore, the partial iterative search which works on different algorithms. It is slightly computationally cheaper, but of course, there are some pitfalls like best subset is not guaranteed right.

So, there is always going to be this potential or missing some good sets or predictors. So, what we actually get is we produce close to best subset. So, that that is something that we

can say that close to a best subsets are definitely we are going to get out of applying partial iterative search.

So, this particular approach is preferred, when we are dealing with large number of predictors right because their the computational time that might be required an exhaustive search might be slightly on the higher side. So, therefore, we would prefer to apply partial iterative search in those situations. So, otherwise if we are just dealing with the moderate number of predictors or going low number of predictors, their exhaustive search is better, because we get a us you know some sort of guarantee of producing the best subset model.

So, therefore, with this between these between these 2 approaches, we can say that there is going to be a trade-off, between computation cost versus potential of finding best subset. So, if we want to you know minimize the computation time, and then probably a partial iterative search is the way to go if we want to you know, if we do not want to compro compromise with the potential of finding best subset, then probably we have we should apply we should employ the exhaustive search.

Now, under this partial iterative search approach we have 3 algorithms that we are going to discuss. So, first one being forward selection so, in the forward selection algorithm.

(Refer Slide Time: 03:00)

The slide is titled "MULTIPLE LINEAR REGRESSION" in bold, dark blue capital letters. Below the title, there is a bulleted list of search algorithms. The first main bullet is "Partial-iterative search algorithms", which has three sub-bullets: "Forward selection" (with two further sub-bullets: "Add predictors one by one" and "Strength as a single predictor is used"), "Backward elimination" (with one sub-bullet: "Drop predictors one by one"), and "Stepwise regression" (with one sub-bullet: "Add predictors one by one and consider dropping insignificant ones"). The second main bullet is "Open RStudio". At the bottom of the slide, there is a dark blue footer bar containing the IIT ROORKEE logo, the text "NPTEL ONLINE CERTIFICATION COURSE", and the page number "19".

- **MULTIPLE LINEAR REGRESSION**
  - Partial-iterative search algorithms
    - Forward selection
      - Add predictors one by one
      - Strength as a single predictor is used
    - Backward elimination
      - Drop predictors one by one
    - Stepwise regression
      - Add predictors one by one and consider dropping insignificant ones
  - Open RStudio

IIT ROORKEE | NPTEL ONLINE CERTIFICATION COURSE | 19

We start with 0 predictors so, we start with no predictor, and we add predictors one by one; so and then the as we go along, the main idea that what happens in forward selection is that strength as a single predictor is actually considered in this approach. So, if a predictor because we are adding one by one, and if it is significant then it would be it would remain in the model, if it is not done it would be excluded from the model. So, therefore, for a particular variable to appear in the in the model to be in the model in a forward selection algorithm, it is strength as a single productor should be on the higher side.

So, that also being the limitation of this forward selection approach. So, we start with no predictors and then we start adding one by one. So, the predictor is significant that it will remain there. Then the second approach is backward elimination. So, in this particular approach we start with all the variables, and then we start dropping them one by one. So, the insignificant mostly insignificant variables are dropped first, and in that order we keep on building the models.

And we keep on dropping till we reach the saturation where all the predictors that are present they are significant. So, this particular approach backward elimination, there is no you know obvious limitation of this approach, except that the computational time that would be requiring this approach would also be slightly on the high higher side in comparison to other partial iterative approaches.

And the third one is a stepwise regression so, in this also we start with just like forward selection approach. So, we start with no predictor and then we add predictors one by one; however, we can consider dropping in significant ones in this particular approach. So, as we move along. So, we keep on adding predictors one by one, and if we if there are some insignificant ones we can consider at any step whether we would like to drop them.

(Refer Slide Time: 05:23)

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

```

89 "Adj.R-sq"=round(mod4sumladjr2,digits=2),
90 mod4sumloutmat[,order(-om)])
91
92 # coefficients of subset models
93 # second argument is index vector indicating ordering in summary output
94 coef(mod4, 1:8)
95
96 # Partial, iterative search
97 # Forward selection
98 mod4=regsubsets(Price~., data = dftrain, nbest = 1, nvmax = NULL,
99 force.in = NULL, force.out = NULL,
100 method = "forward", intercept = T)
101 mod4sum=summary(mod4)
102
103 om1=as.integer(apply(mod4sumloutmat, 2, countspsch)); om1
104 data.frame("#Coeffs"=as.integer(apply(mod4sumloutmat, 1, countspsch)),

```

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

```

Console
C:/Users/user/Desktop/MOOC January 2018/Dr. Gaurav Dixit/Session 6/ >
(Intercept) Fuel_typePetrol SR_Price Age
4.0676722 -1.0549172 0.2638007 -0.3682494

[[4]]
(Intercept) Fuel_typePetrol SR_Price KM Age
4.56626891 -1.07571364 0.26705794 -0.01122816 -0.30720072

[[5]]
(Intercept) Fuel_typePetrol SR_Price KM Owners
4.1971710 -1.1809765 0.2706734 -0.0127651 0.5188228
Age
-0.3208142

```

The environment pane shows the following objects:

- `dftrain`: 45 obs. of 8 variables
- `mod4`: List of 28
- `mod4sum`: List of 8
- `om`: int [1:8] 1 6 8 5 2 4 3 7

The right pane shows the documentation for the `regsubsets` function, including its description and usage.

So, these are 3 main approaches. So, let us understand them a bit more using an exercise. So, a partial iterative search first we are going to start with the forward selection approach. So, in this as you can see the data set that we are going to use is the same; that is, a used car data set so, this is pre-loaded partitioning is already done, and we are dealing with the what number observation that we have is 75 up to excluding all the outlets that are there, and the variables often trust only 8 variables are there, ba one being the price that is outcome variable of interest. So, again you can see that we are using this dead subsets function, and the price and then this formula is specified as price tilde dot that includes all of the predictors in the data set. Now you can see the next important argument data is mentioned as d f train and then method is forward.

So, that tells the function that we would like to apply forward selection algorithm. So, let us execute this code.

(Refer Slide Time: 06:27)

```

89 "Adj.R-sq"=round(mod3sumladjr2,digits=2),
90 mod3sumloutmat[,order(-om)])
91
92 # coefficients of subset models
93 # second argument is index vector indicating ordering in summary output
94 coef(mod3, 1:8)
95
96 # Partial, iterative search
97 # Forward selection
98 mod4=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
99 force.in = NULL, force.out = NULL,
100 method = "forward", intercept = T)
101 mod4sum=summary(mod4)
102
103 om1=as.integer(apply(mod4sum$outmat, 2, countspsch)); om1
104 data.frame("#Coeffs"=as.integer(apply(mod4sum$outmat, 1, countspsch)),
105

```

```

0.53760074 -0.24389649 -0.33418197

[[8]]
(Intercept) Fuel_typeDiesel Fuel_typePetrol SR_Price KM
4.33415354 -0.07259189 -1.29888125 0.28150429 -0.01342220
Transmission1 Owners Airbag Age
0.07071214 0.54039808 -0.24420109 -0.33561279

> mod4=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
+ force.in = NULL, force.out = NULL,
+ method = "forward", intercept = T)
+

```

Now, let us generate the summary of this result this particular model. Now as we did in the previous lecture that we would like to count the special character that is asterisk in this case. So, that the particular function count is special character is already created here you can see here for the special character. So, we can use this particular function, and count the number of asterisk, number of asterisk that are there in this particular matrix for different columns. So, this particular result we are going to as we discussed in the previous lecture, this particular result is going to be used as an index vector for us later on.

(Refer Slide Time: 07:13)

```

97 # Forward selection
98 mod4=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
99 force.in = NULL, force.out = NULL,
100 method = "forward", intercept = T)
101 mod4sum=summary(mod4)
102
103 om1=as.integer(apply(mod4sum$outmat, 2, countspsch)); om1
104 data.frame("#Coeffs"=as.integer(apply(mod4sum$outmat, 1, countspsch)),
105
106 "cp"=round(mod4sum$cp, digits=2),
107 "R-sq"=round(mod4sum$rsq, digits=2),
108 "Adj.R-sq"=round(mod4sum$adjr2,digits=2),
109 mod4sumloutmat[,order(-om1)])
110
111 coef(mod4, 1:8)
112
113 # Backward elimination

```

```

(Intercept) Fuel_typeDiesel Fuel_typePetrol SR_Price KM
4.33415354 -0.07259189 -1.29888125 0.28150429 -0.01342220
Transmission1 Owners Airbag Age
0.07071214 0.54039808 -0.24420109 -0.33561279

> mod4=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
+ force.in = NULL, force.out = NULL,
+ method = "forward", intercept = T)
> mod4sum=summary(mod4)
> om1=as.integer(apply(mod4sum$outmat, 2, countspsch)); om1
[1] 1 6 8 5 2 4 3 7

```

(Refer Slide Time: 07:49)

The screenshot shows the RStudio interface with a script editor on the left and the Environment pane on the right. The script defines a function `stepAIC` that performs model selection using AIC. The function iterates over a set of models, selecting the one with the lowest AIC. The results are stored in the `mod4sum` object.

```

# stepAIC
stepAIC = function(x, y, data, weights=NULL, nbest=1, mvmax=NULL,
  force.in=NULL, force.out=NULL, method="forward", intercept=T) {
  mod4sum=summary(mod4)
  on1=as.integer(apply(mod4sum$outmat, 2, countsppch)); on1
  data.frame("Ccoeffs"=as.integer(apply(mod4sum$outmat, 1, countsppch)),
    "RSS"=mod4sum$rrss,
    "Cp"=round(mod4sum$cpc, digits=2),
    "R-sq"=round(mod4sum$rsq, digits=2),
    "Adj-R-sq"=round(mod4sum$adjr2, digits=2),
    mod4sum$outmat[,order("on1")])
  coef(mod4, 1:8)
  ## backward elimination
  mod4$regsubsets(Price~., data = dtrain, nbest = 1, mvmax = NULL,
    lty=c(1:nv))
}

```

The Environment pane shows the results of the function, including the best model and its coefficients.

```

mod4sum List of 8
on1 List of 28
mod4sum List of 8
on1 int [1:8] 1 6 8 5 2 4 3 7
on1 int [1:8] 1 6 8 5 2 4 3 7
partidx int [1:45] 42 66 5 36 1 ..

```

The Console shows the output of the function, including the best model and its coefficients.

```

# Console
C:\Users\user\Desktop\MOOC January 2018\Dr. Gaurav Dhill/Gaonkar R\
+ mod4sum$outmat[,order("on1")]
x.Coeffs RSS Cp R.sq Adj.R.sq SR_Price Age_Fuel_typePetrol KM
1 (1) 1.71.56760 32.29 0.49 0.48 * * *
2 (1) 2.477847 9.94 0.66 0.64 * * *
3 (1) 3.37.81514 1.73 0.73 0.71 * * *
4 (1) 4.36.12397 1.99 0.74 0.72 * * *
5 (1) 5.35.36094 3.21 0.75 0.71 * * *
6 (1) 6.35.18017 5.03 0.75 0.71 * * *
7 (1) 7.35.15838 7.00 0.75 0.70 * * *
8 (1) 8.35.15357 9.00 0.75 0.69 * * *
Owners Airbag Transmission1 Fuel_typeDiesel

```

So, 4 variable model is the model that we have to select if we follow the criteria of adjusted R square. If we look at the R square value, this is also a very similar to or is exactly same as what we had an exhaustive search 0.49, 0.66, 0.73 and then finally, reaching 0.74 and then 0.75. So, same result and the numbers are also same for c p value though we had the, we have with us the results of exhaustive search. So, you can see

these are this is this particular table is the results of exhaustive search, and you can see the same numbers are there the output that we have got from forward selection are happens to be happens to be same for forward selection. So, this is just in this in this particular case the data set that we have and the partitioning that we perform. If we change the partitioning the results might also change and also what we are dealing with a small data set. So, therefore, we might not see much difference between these 2 approaches.

Now, if we look at the variables, you can see show room price a surface being present in the all 8 models, starting from one variable model to 8 variable models and all of them it is present then age is present in same in or such models starting from 2 variable models to 8 models. So, early fuel type petrol is there then k m is there. And then comes the owners and other variables that are there.

(Refer Slide Time: 10:05)

```

98 mod4=regsubsets(price~., data = dtrain, nbest = 1, nvmax = NULL,
99               force.in = NULL, force.out = NULL,
100              method = "forward", intercept = T)
101 mod4sum=summary(mod4)
102
103 om1=as.integer(apply(mod4sum$outmat, 2, countspch)); om1
104 data.frame("#coeffs"=as.integer(apply(mod4sum$outmat, 1, countspch)),
105           "RSS"=mod4sum$rss,
106           "Cp"=round(mod4sum$cp, digits=2),
107           "R-sq"=round(mod4sum$rsq, digits=2),
108           "Adj R-sq"=round(mod4sum$adjr2, digits=2),
109           mod4sum$outmat[,order(-om1)])
110
111 coef(mod4, 1:8)
112
113 # backward elimination
114 mod5=regsubsets(price~., data = dtrain, nbest = 1, nvmax = NULL,
115               method = "backward", intercept = T)

```

Console Output:

```

7 ( 1) 7 35.15838 7.00 0.75 0.70 * *
8 ( 1) 8 35.15357 9.00 0.75 0.69 * *

```

Owners Airbag Transmission Fuel type Diesel

```

1 ( 1)
2 ( 1)
3 ( 1)
4 ( 1)
5 ( 1)
6 ( 1)
7 ( 1)
8 ( 1)

```

Environment History

- mod3sum List of 8
- mod4 List of 28
- mod4sum List of 8
- om1 int [1:8] 1 6 8 5 2 4 3 7
- om1 int [1:8] 1 6 8 5 2 4 3 7
- partidx int [1:45] 43 2 66 5 36 1 ..

Functions

Files Plots Packages Help Viewer

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```

regsubsets(x[, ...])

```

## S3 method for class 'formula'

```

regsubsets(v, data, weights=NULL, nbest
  , force.in=NULL, force.out=NULL, intercept=

```

So, if we are interested in understanding the coefficient. So, let us run this particular function coefficient and pricing passing the model argument we will get this.



(Refer Slide Time: 10:19)

```

100 mod4summary(mod4)
101
102 mod4summary(mod4)
103
104 on1=as.integer(apply(mod4summat, 2, countspsch)); on1
105 data.frame("#coeffs"=as.integer(apply(mod4summat, 1, countspsch)),
106           "RSS"=mod4sumriss,
107           "Cp"=round(mod4sumicp, digits=2),
108           "R-sq"=round(mod4sumirsq, digits=2),
109           "Adj-R-sq"=round(mod4sumiadjr2, digits=2),
110           mod4summat[,order(-on1)])
111
112 coef(mod4, 1:8)
113
114 # Backward elimination
115 mod5=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
116               force.in = NULL, force.out = NULL,
117               method = "backward", intercept = T)
118
119 console
120 C:\Users\user\Desktop\MOOC January 2018\Dr. Gaurav Dhill\Session 6\
121
122 (Intercept) Fuel_typePetrol SR_Price KM Transmission1
123 4.26476158 -1.22800167 0.28129203 -0.01347426 0.06897438
124 Owners Airbag Age
125 0.53760074 -0.24389649 -0.33418197
126
127 [[8]]
128 (Intercept) Fuel_typeDiesel Fuel_typePetrol SR_Price KM
129 4.33415354 -0.07259189 -1.29888125 0.28150429 -0.01342220
130 Transmission1 Owners Airbag Age
131 0.07071214 0.54039808 -0.24420109 -0.33561279
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, you can see the one variable model the coefficient comes out to be 0.24 5 9 5 this is the value.

(Refer Slide Time: 10:25)

```

100 mod4summary(mod4)
101
102 mod4summary(mod4)
103
104 on1=as.integer(apply(mod4summat, 2, countspsch)); on1
105 data.frame("#coeffs"=as.integer(apply(mod4summat, 1, countspsch)),
106           "RSS"=mod4sumriss,
107           "Cp"=round(mod4sumicp, digits=2),
108           "R-sq"=round(mod4sumirsq, digits=2),
109           "Adj-R-sq"=round(mod4sumiadjr2, digits=2),
110           mod4summat[,order(-on1)])
111
112 coef(mod4, 1:8)
113
114 # Backward elimination
115 mod5=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
116               force.in = NULL, force.out = NULL,
117               method = "backward", intercept = T)
118
119 console
120 C:\Users\user\Desktop\MOOC January 2018\Dr. Gaurav Dhill\Session 6\
121
122 (Intercept) Fuel_typePetrol SR_Price KM Transmission1
123 4.26476158 -1.22800167 0.28129203 -0.01347426 0.06897438
124 Owners Airbag Age
125 0.53760074 -0.24389649 -0.33418197
126
127 [[8]]
128 (Intercept) Fuel_typeDiesel Fuel_typePetrol SR_Price KM
129 4.33415354 -0.07259189 -1.29888125 0.28150429 -0.01342220
130 Transmission1 Owners Airbag Age
131 0.07071214 0.54039808 -0.24420109 -0.33561279
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, s R price being the only variable, then we look at the 2-variable model, we have a s R price and age then you can see age is negatively related here.



(Refer Slide Time: 10:39)

```

100  mod4sumsummary(mod4)
101  method = "forward", intercept = 1)
102
103  om1=as.integer(apply(mod4sumoutmat, 2, countspch)); om1
104  data.frame("#coeffs"=as.integer(apply(mod4sumoutmat, 1, countspch)),
105            "RSS"=mod4sumlrss,
106            "Cp"=round(mod4sumlcp, digits=2),
107            "R-sq"=round(mod4sumlrsq, digits=2),
108            "Adj R-sq"=round(mod4sumladr2, digits=2),
109            mod4sumoutmat[,order(-om1)])
110
111  coef(mod4, 1.8)
112
113  # Backward elimination
114  mod5=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
115                force.in = NULL, force.out = NULL,
116                method = "backward", intercept = 1)
117
118  (top1new) 2
  
```

Console Output:

```

[[2]]
(Intercept)  SR_Price  Age
3.5977612    0.2802512  -0.3656525

[[3]]
(Intercept) Fuel_typePetrol  SR_Price  Age
4.0676722    -1.0549172    0.2638007  -0.3682494

[[4]]
(Intercept) Fuel_typePetrol  SR_Price  KM  Age
4.56626891    -1.07571364    0.26705794  -0.01122816  -0.30720072
  
```

Environment:

- Global Environment
- mod3sum: List of 8
- mod4: List of 28
- mod5sum: List of 8
- om: int [1:8] 1 6 8 5 2 4 3 7
- om1: int [1:8] 1 6 8 5 2 4 3 7
- partidx: int [1:45] 43 2 66 5 36 1 ..

Functions:

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(x[, ...])
```

## S3 method for class 'loommin'

```
regsubsets(x[, data, weights=NULL, nbest=1, force.in=NULL, force.out=NULL, method="backward", intercept=1])
```

And then fuel type petrol and s R price age. So, just like in the exhaustive search. So, we are getting very same results the same models also. Then after fuel type petrol s R price we see the kilometers appearing in their. So, it is happening exactly like what happened in the exhaustive search right.

However, as I said if we change the sample of or if we change the partitioning, we increase the sample size already in change the partitioning I read on the redo the partitioning then probably the results will change. Now let us come to our next algorithm that is backward elimination. So, again for backward elimination as well we can use the same function less subsets.

And you can see the one change that we have done here is the in the method argument where we have a specified backward as the algorithm in this case. So, let us execute this code, and let us also generate the summary of this output. Now here again we are going to do the same thing that counting of a special character and the other things remain same. So, let us execute this code ok.

(Refer Slide Time: 12:00)

```

114 mod5<-regsubsets(Price~., data = dftrain, nbest = 1, nvmax = NULL,
115 force.in = NULL, force.out = NULL,
116 method = "backward", intercept = T)
117 mod5sum<-summary(mod5)
118
119 om2<-as.integer(apply(mod5sum$outmat, 2, countspch)); om2
120 data.frame("#Coeffs"=as.integer(apply(mod5sum$outmat, 1, countspch)),
121 "RSS"=mod5sum$rss,
122 "Cp"=round(mod5sum$cp, digits=2),
123 "R-sq"=round(mod5sum$rsq, digits=2),
124 "Adj. R-sq"=round(mod5sum$adjr2, digits=2),
125 mod5sum$outmat[,order(-om2)])
126
127 coef(mod5, 1:8)
128
129 # Sequential replacement
130 mod6<-regsubsets(Price~., data = dftrain, nbest = 1, nvmax = NULL,
131 method = "backward", intercept = T)

```

Console:

```

> mod5<-regsubsets(Price~., data = dftrain, nbest = 1, nvmax = NULL,
+ force.in = NULL, force.out = NULL,
+ method = "backward", intercept = T)
> mod5sum<-summary(mod5)
+ data.frame("#Coeffs"=as.integer(apply(mod5sum$outmat, 1, countspch)),
+ "RSS"=mod5sum$rss,
+ "Cp"=round(mod5sum$cp, digits=2),
+ "R-sq"=round(mod5sum$rsq, digits=2),
+ "Adj. R-sq"=round(mod5sum$adjr2, digits=2),
+ mod5sum$outmat[,order(-om2)])
Error in order(-om2) : object 'om2' not found

```

Environment:

- mod5sum: List of 8
- mod4: List of 28
- mod4sum: List of 8
- mod5: List of 28
- mod5sum: List of 8
- om: int [1:8] 1 6 8 5 2 4 3 7
- om1: int [1:8] 1 6 8 5 2 4 3 7

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(xw, ...)
```

## 23 method for class 'lqsmult'

Oh, we did not create O m 2. So, let us compute this, and then produce this data frame.

(Refer Slide Time: 12:10)

```

114 mod5<-regsubsets(Price~., data = dftrain, nbest = 1, nvmax = NULL,
115 force.in = NULL, force.out = NULL,
116 method = "backward", intercept = T)
117 mod5sum<-summary(mod5)
118
119 om2<-as.integer(apply(mod5sum$outmat, 2, countspch)); om2
120 data.frame("#Coeffs"=as.integer(apply(mod5sum$outmat, 1, countspch)),
121 "RSS"=mod5sum$rss,
122 "Cp"=round(mod5sum$cp, digits=2),
123 "R-sq"=round(mod5sum$rsq, digits=2),
124 "Adj. R-sq"=round(mod5sum$adjr2, digits=2),
125 mod5sum$outmat[,order(-om2)])
126
127 coef(mod5, 1:8)
128
129 # Sequential replacement
130 mod6<-regsubsets(Price~., data = dftrain, nbest = 1, nvmax = NULL,
131 method = "backward", intercept = T)

```

Console:

```

+ "Adj. R-sq"=round(mod5sum$adjr2, digits=2),
+ mod5sum$outmat[,order(-om2)])
+
+ x Coeffs RSS Cp R.sq Adj. R.sq SR_Price Age Fuel_typePetrol KM
1 ( 1 ) 1 71.56760 32.29 0.49 0.48 *
2 ( 1 ) 2 47.78747 9.94 0.66 0.64 *
3 ( 1 ) 3 37.81514 1.73 0.73 0.71 *
4 ( 1 ) 4 36.12397 1.99 0.74 0.72 *
5 ( 1 ) 5 35.36094 3.21 0.75 0.71 *
6 ( 1 ) 6 35.18017 5.03 0.75 0.71 *
7 ( 1 ) 7 35.15838 7.00 0.75 0.70 *
8 ( 1 ) 8 35.15357 9.00 0.75 0.69 *
Owners Airbag Transmission Fuel_typeDiesel
1 ( 1 )

```

Environment:

- mod4: List of 28
- mod4sum: List of 8
- mod5: List of 28
- mod5sum: List of 8
- om: int [1:8] 1 6 8 5 2 4 3 7
- om1: int [1:8] 1 6 8 5 2 4 3 7
- om2: int [1:8] 1 6 8 5 2 4 3 7

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(xw, ...)
```

## 23 method for class 'lqsmult'

Now, again if we look at the results of backward elimination, then also we would see that same numbers are there again. So, adjusted R square value again starting from the same 0.48 0.64. So, the values these values for all 3 algorithm whether exhaustive or forward selection our backward simulation what we are saying is up to 2 decimal points. So, there might be some difference may be there may be some difference if we look at maybe up to 6 decimal points. So, the adjusted R square column you can see that speaking at 4

variable model 0.72 is the value. similarly, for R square also it is peaking at the you know 5 variable model 0.75 is the value. And the mellow c p is the numbers are same. So, we will have to select again 3 variable model, but the value of c p is one 0.73.

So, in this so, the R s s residual sum of squares number, there also seems to be same. Let us look at the variables you can see again a s R price present in all 8 models, and age is present and the 7 models followed by fuel type petrol present in 6 models, and followed by kilometer k m which is present in 5 models. So, if we have to we want to look at the coefficients for different variable models, we can you do the same using coefficient function.

(Refer Slide Time: 13:42)

The screenshot shows an RStudio window with a script editor, a console, and a sidebar. The script editor contains R code for model selection using the `stepAIC` function. The console shows the output of the code, including the results of `coef(mod5, 1:8)` and `mod5regsubsets`. The sidebar shows the 'Environment' pane with a list of objects and the 'Help' pane with documentation for `functions for model selection`.

```

118 mod5summary(mod5)
119 mod5$summary(mod5)
120 data.frame("#coeffs"=as.integer(apply(mod5$summary, 1, countspch)),
121            "RSS"=mod5$summary$RSS,
122            "Cp"=round(mod5$summary$Cp, digits=2),
123            "R-sq"=round(mod5$summary$Rsq, digits=2),
124            "Adj R-sq"=round(mod5$summary$AdjRsq, digits=2),
125            mod5$summary$outmat[,order(-om2)])
126
127 coef(mod5, 1:8)
128
129 # Sequential replacement
130 mod5regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
131               force.in = NULL, force.out = NULL,
132               method = "stepAIC", intercept = T)

```

Console output:

```

> coef(mod5, 1:8)
[[1]]
(Intercept)    SR_Price
1.7981689    0.2459522

[[2]]
(Intercept)    SR_Price    Age
3.5977612    0.2802512   -0.3656525

```

Environment pane:

- mod4: List of 28
- mod4summary: List of 8
- mod5: List of 28
- mod5summary: List of 8
- om: int [1:8] 1 6 8 5 2 4 3 7
- om1: int [1:8] 1 6 8 5 2 4 3 7
- om2: int [1:8] 1 6 8 5 2 4 3 7

Help pane: functions for model selection

Description: Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage: `regsubsets(x[, ...])`

So, let us look at this again you would see that s R price is present in the first model and then 6 model you can look at the coefficient also age is again negatively correlated as expected in the 2-variable model, then in 3 variable model again we have the fuel entry of fuel type petrol with the negative coefficient negative regression coefficient. And then you would see in the 4-variable model, we see the k m as well with the negative regression coefficient here.

In the same fashion the results again seemed to be very similar. So, let us move to our next algorithm, that is sequential replacement. So, for sequential replacement there is something that we have not discussed.

(Refer Slide Time: 14:29)

```

119 om2<-as.integer(apply(mod5sum$outmat, 4, countspsch)); om2
120 data.frame("Ccoeffs"-as.integer(apply(mod5sum$outmat, 1, countspsch)),
121           "RSS"-mod5sum$rss,
122           "Cp"-round(mod5sum$cp, digits=2),
123           "R-sq"-round(mod5sum$rsq, digits=2),
124           "Adj.R-sq"-round(mod5sum$adjr2, digits=2),
125           mod5sum$outmat[,order(-om2)])
126
127 coef(mod5, 1:8)
128
129 # Sequential replacement
130 mod6<-regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
131                 force.in = NULL, force.out = NULL,
132                 method = "seqrep", intercept = T)
133 mod6sum<-summary(mod6)
134
135 om3<-as.integer(apply(mod6sum$outmat, 2, countspsch)); om3

```

Console

```

[1] 4.56626891 -1.07571364 0.26705794 -0.01122816 -0.30720072
[[5]]
(Intercept) Fuel_typePetro1 SR_Price KM Age Owners
4.1971710 -1.1809765 0.2706734 -0.0127651 0.5188228
-0.3208142
[[6]]
(Intercept) Fuel_typePetro1 SR_Price KM Age Owners
4.24072267 -1.20938318 0.28263429 -0.01345213 0.53360580
Airhan Age

```

Environment History

- mod4 List of 28
- mod4sum List of 8
- mod5 List of 28
- mod5sum List of 8
- om int [1:8] 1 6 8 5 2 4 3 7
- om1 int [1:8] 1 6 8 5 2 4 3 7
- om2 int [1:8] 1 6 8 5 2 4 3 7

Functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```

regsubsets(x, ...)

```

## S3 method for class 'loammin'

```

regsubsets(x, data, weights=NULL, nbest=
  1000, adjRsq=TRUE, force.in=NULL, force.out=NULL,

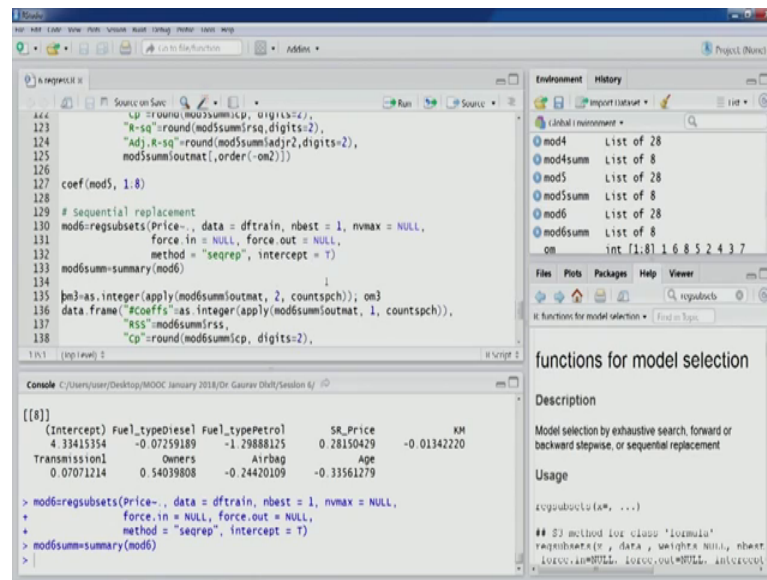
```

So, what happens sequential replacement is or we start with all the variables, we start with all the variables, and from there then we select you know, and then we identify the best subsets from their best you know subset model that would be there. Then for each of for each of the variable in the in the in that subset for each variable we try a different whether that can be replaced by some other variable right.

So, in that sense we start out, and then these so, in the subset model that is selected for each variable will try out for it is replacement. And then again, we will have for if there are 4 variables in the subset model, and for each variable we will try out different other variables as a as a replacement. And therefore, we will get 4 more models, and out of those 4 models, will again check the, which one is performing better if it happens. Then if then we select the best one, and then we proceed further the same thing is applied again and again till we are not able to find the best model.

So, this is what we call sequential replacement algorithm. And the stepwise regression approach would also be very similar to what is we specified as you can see the replacement.

(Refer Slide Time: 15:56)



```
##
123   "R-sq"=round(modsum$rsq,digits=2),
124   "Adj.R-sq"=round(modsum$adjr2,digits=2),
125   modsum$outmat[,order(-om2)])
126
127 coef(mod5, 1:8)
128
129 # sequential replacement
130 mod6=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
131               force.in = NULL, force.out = NULL,
132               method = "seqrep", intercept = T)
133 modsum=summary(mod6)
134
135 m3=as.integer(apply(modsum$outmat, 2, countspch)); om3
136 data.frame("Rcoeffs"=as.integer(apply(modsum$outmat, 1, countspch)),
137           "RSS"=modsum$rss,
138           "Cp"=round(modsum$cp, digits=2),
```

```
[[8]]
      (Intercept) Fuel_typeDiesel Fuel_typePetrol      SR_Price      KM
1  4.33415354    -0.07259189    -1.29888125    0.28150429    -0.01342220
2  0.07071214    0.54039808    -0.24420109    -0.33561279
3  0.07071214    0.54039808    -0.24420109    -0.33561279
4  0.07071214    0.54039808    -0.24420109    -0.33561279
5  0.07071214    0.54039808    -0.24420109    -0.33561279
6  0.07071214    0.54039808    -0.24420109    -0.33561279
7  0.07071214    0.54039808    -0.24420109    -0.33561279
8  0.07071214    0.54039808    -0.24420109    -0.33561279

> mod6=regsubsets(Price~., data = dtrain, nbest = 1, nvmax = NULL,
+               force.in = NULL, force.out = NULL,
+               method = "Seqrep", intercept = T)
> modsum=summary(mod6)
>
```

Environment History

- mod4 List of 28
- mod4sum List of 8
- mod5 List of 28
- mod5sum List of 8
- mod6 List of 28
- mod6sum List of 8
- om Int [1,8] 1 6 8 5 2 4 3 7

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```
regsubsets(x, ...)
```

```
## S3 method for class 'lmAmin'
regsubsets(x, data, weights=NULL, nbest=
  10000,adjust=TRUE, force.in=NULL, force.out=NULL, intercept=
```

Some variations might differ, then with the name also, there are going to be very variation depending on the implementation that we follow. So, in this case as you can see we are again using the reg subset function. And the one difference that we can see is the method sequence replacement s e q R e p has been specified. So, let us execute this , it is run this model.

Let us also compute this number of this asterisk in each column of the output matrix. Let us compute this matrix. Now we can see again, in the adjusted R square value again it is speaking at the 4-variable model, and the same numbers are there for R square also speaking at the 5-variable model, and the same numbers are there again for cp also 3 variable is seems to be the more the appropriate one and the same numbers for R s s as well.

(Refer Slide Time: 16:56)

```

130 mod0=regsubsets(price~., data = dftrain, nbest = 1, nvmax = NULL,
131               force.in = NULL, force.out = NULL,
132               method = "stepAIC", intercept = T)
133 mod6sum=summary(mod6)
134
135 om3=as.integer(apply(mod6sum$outmat, 2, countspsch)); om3
136 data.frame("#Coeffs"=as.integer(apply(mod6sum$outmat, 1, countspsch)),
137           "RSS"=mod6sum$rss,
138           "Cp"=round(mod6sum$cp, digits=2),
139           "R-sq"=round(mod6sum$rsq, digits=2),
140           "Adj. R-sq"=round(mod6sum$adjr2, digits=2),
141           mod6sum$outmat[,order(-om3)])
142
143 coef(mod6, 1:8)
144
145 # Stepwise regression
146 mod7=stepAIC(price~., data = dftrain, direction = "both")
147
148

```

Console Output:

	X	coeffs	RSS	Cp	R.sq	Adj. R.sq	SR	Price	Age	Fuel_type	Petrol	KM
1	( 1 )		171.56760	32.29	0.49	0.48	*					
2	( 1 )		247.78747	9.94	0.66	0.64	*	*				
3	( 1 )		337.81514	1.73	0.73	0.71	*	*	*			
4	( 1 )		436.12397	1.99	0.74	0.72	*	*	*	*		
5	( 1 )		535.36094	3.21	0.75	0.71	*	*	*	*	*	
6	( 1 )		635.18017	5.03	0.75	0.71	*	*	*	*	*	*
7	( 1 )		735.15838	7.00	0.75	0.70	*	*	*	*	*	*
8	( 1 )		835.15357	9.00	0.75	0.69	*	*	*	*	*	*

Owners Airbag Transmission Fuel\_type Diesel

For the variables also, a s R price being presented in all the models then followed by age and then fuel type petrol and came the result seem to be exactly same.

(Refer Slide Time: 17:09)

```

130 mod0=regsubsets(price~., data = dftrain, nbest = 1, nvmax = NULL,
131               force.in = NULL, force.out = NULL,
132               method = "stepAIC", intercept = T)
133 mod6sum=summary(mod6)
134
135 om3=as.integer(apply(mod6sum$outmat, 2, countspsch)); om3
136 data.frame("#Coeffs"=as.integer(apply(mod6sum$outmat, 1, countspsch)),
137           "RSS"=mod6sum$rss,
138           "Cp"=round(mod6sum$cp, digits=2),
139           "R-sq"=round(mod6sum$rsq, digits=2),
140           "Adj. R-sq"=round(mod6sum$adjr2, digits=2),
141           mod6sum$outmat[,order(-om3)])
142
143 coef(mod6, 1:8)
144
145 # Stepwise regression
146 mod7=stepAIC(price~., data = dftrain, direction = "both")
147
148

```

Console Output:

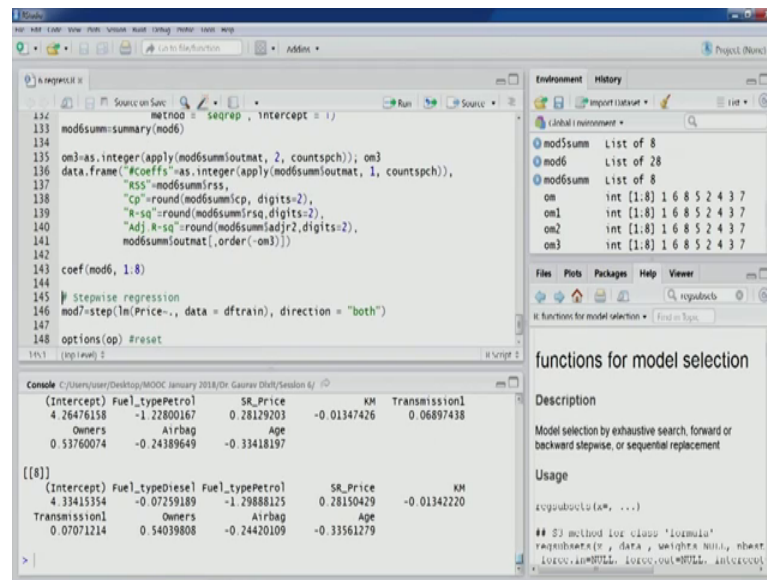
	X	coeffs	RSS	Cp	R.sq	Adj. R.sq	SR	Price	Age	Fuel_type	Petrol	KM
7	( 1 )		735.15838	7.00	0.75	0.70	*	*	*	*	*	*
8	( 1 )		835.15357	9.00	0.75	0.69	*	*	*	*	*	*

Owners Airbag Transmission Fuel\_type Diesel

Now, we are interested in looking at the coefficient value, then we can do.



(Refer Slide Time: 17:19)



```

152 mod6sum=summary(mod6)
153
154 om3=as.integer(apply(mod6sum$outmat, 2, countspch)); om3
155 data.frame("#coeffs"=as.integer(apply(mod6sum$outmat, 1, countspch)),
156           "RSS"=mod6sum$rss,
157           "Cp"=round(mod6sum$cp, digits=2),
158           "R-sq"=round(mod6sum$rsq, digits=2),
159           "Adj R-sq"=round(mod6sum$adjr2, digits=2),
160           mod6sum$outmat[,order(-om3)])
161
162 coef(mod6, 1:8)
163
164
165 # Stepwise regression
166 mod7=stepAIC(mod6, data = dtrain, direction = "both")
167
168 options(op) #reset
169
170

```

Environment

- mod6sum List of 8
- mod6 List of 28
- mod6sum List of 8
- om int [1:8] 1 6 8 5 2 4 3 7
- om1 int [1:8] 1 6 8 5 2 4 3 7
- om2 int [1:8] 1 6 8 5 2 4 3 7
- om3 int [1:8] 1 6 8 5 2 4 3 7

functions for model selection

Description

Model selection by exhaustive search, forward or backward stepwise, or sequential replacement

Usage

```

stepAIC(x, ...)

```

## S3 method for class 'lmAIC'

```

stepAIC(x, data, weights=NULL, subset

```

So, using coefficient function and again as we can see this, again the same results are there is no change s R price then s R price and age, then fuel type petrol s R price and age, and then fuel type petrol s R price and then entry of kilometer k m and the same fashion, the same result we get. Stepwise regression there is another function that is available to us for stepwise func regression that is called step.

So, in this case and this is a step function we have to pass on the again we have to pass on the l m function as the first argument. So, in the l m function we would be specifying the formula as usual in the first argument and then the data. The direction is specified as both so, we can add and remove. So, both the kind of operation can be performed. If you are understand in finding more about this particular function, you can go into the help section type step and choose them.



(Refer Slide Time: 18:16)

```

132 method = "stepAIC", intercept = 1)
133 mod6sum=summary(mod6)
134
135 om3=as.integer(apply(mod6sum$outmat, 2, countspch)); om3
136 data.frame("#coeffs"=as.integer(apply(mod6sum$outmat, 1, countspch)),
137           "RSS"=mod6sum$rss,
138           "Cp"=round(mod6sum$cp, digits=2),
139           "R-sq"=round(mod6sum$rsq, digits=2),
140           "Adj. R-sq"=round(mod6sum$adjr2, digits=2),
141           mod6sum$outmat[,order(-om3)])
142
143 coef(mod6, 1:8)
144
145 # Stepwise regression
146 mod7=stepAIC(data = dftrain, direction = "both")
147
148 options(op) #reset

```

Console output:

```

[[4]]
(Intercept) Fuel_typePetro1    SR_Price    KM    Age
4.56626891   -1.07571364    0.26705794   -0.01122816   -0.30720072

[[5]]
(Intercept) Fuel_typePetro1    SR_Price    KM    Owners
4.1971710   -1.1809765    0.2706734   -0.0127651    0.5188228
Age
-0.3208142

[[6]]

```

So, you can see the choose a model a i c in a stepwise algorithm.

So, you can see different options are there this particular method would also allow us to run backward forward and then both.

(Refer Slide Time: 18:33)

```

132 method = "stepAIC", intercept = 1)
133 mod6sum=summary(mod6)
134
135 om3=as.integer(apply(mod6sum$outmat, 2, countspch)); om3
136 data.frame("#coeffs"=as.integer(apply(mod6sum$outmat, 1, countspch)),
137           "RSS"=mod6sum$rss,
138           "Cp"=round(mod6sum$cp, digits=2),
139           "R-sq"=round(mod6sum$rsq, digits=2),
140           "Adj. R-sq"=round(mod6sum$adjr2, digits=2),
141           mod6sum$outmat[,order(-om3)])
142
143 coef(mod6, 1:8)
144
145 # Stepwise regression
146 mod7=stepAIC(data = dftrain, direction = "both")
147
148 options(op) #reset

```

Console output:

```

[[4]]
(Intercept) Fuel_typePetro1    SR_Price    KM    Age
4.56626891   -1.07571364    0.26705794   -0.01122816   -0.30720072

[[5]]
(Intercept) Fuel_typePetro1    SR_Price    KM    Owners
4.1971710   -1.1809765    0.2706734   -0.0127651    0.5188228
Age
-0.3208142

[[6]]

```

So, as we discussed that the stepwise regression is about a star just like forward regression, and at each step we can consider dropping off insignificant variables. So, that is why the both indicating meaning the same both indicating the step wise regression. And the other options using the same function we can also, build backward and forward

as well which we had just done using read subsets. So, in this particular function the a i c is used to find out the different subset models. For more information you can always look at the other arguments that are there. So, let us use this function so, let us compute this.

(Refer Slide Time: 19:30)

```

133 mod5sum<-summary(mod5)
134
135 om3<-as.integer(apply(mod5sum$outmat, 2, countspsch)); om3
136 data.frame("Coeffs"=as.integer(apply(mod5sum$outmat, 1, countspsch)),
137           "RSS"=mod5sum$rss,
138           "Cp"=round(mod5sum$cp, digits=2),
139           "R-sq"=round(mod5sum$rsq, digits=2),
140           "Adj R-sq"=round(mod5sum$adjr2, digits=2),
141           mod5sum$outmat[,order(-om3)])
142
143 coef(mod6, 1:8)
144
145 # Stepwise regression
146 mod7<-stepAIC(mod7=lm, data = dftrain, direction = "both")
147
148 options(op) #reset
149

```

Price	Fuel_type	SR_Price	KM	Age	AIC
-none>					36.124 2.113
- KM			1		1.673 37.796 2.150
+ Owners			1		0.767 35.356 3.147
+ Airbag			1		0.139 35.985 3.939
+ Transmission			1		0.002 36.122 4.111
- Fuel_type			2		10.346 46.470 9.446
- Age			1	11.553	47.677 12.601
- SR_Price		1	72.559	108.683	49.680

And this is the results that you get. So, you can see that we start with a i c value up 6.89, and the formula that we start with is price and then other variables, fuel type s R price k m transmission owners and airbag plus age. So, you can see all 7 variables are present, and we start with a i c value of 6.89.

Now if we look at the possible additions, and possible addition, or elimination that that could be there. You can see that if we eliminate transmission, we would gain will reach the a i c level of 4.917, and that would be and that would be a lower a i c value, and then followed by heavy remove airbag.

(Refer Slide Time: 20:23)

```

133 modSumm=summary(mod6)
134
135 om3=as.integer(apply(modSumm$outmat, 2, countspch)); om3
136 data.frame("#Coeffs"=as.integer(apply(modSumm$outmat, 1, countspch)),
137           "RSS"=modSumm$rss,
138           "Cp"=round(modSumm$cp, digits=2),
139           "R-sq"=round(modSumm$rsq, digits=2),
140           "Adj. R-sq"=round(modSumm$adjr2, digits=2),
141           modSumm$outmat[,order(-om3)])
142
143 coef(mod6, 1:8)
144
145 # Stepwise regression
146 mod7=stepAIC(mod6, data = dftrain, direction = "both")
147
148 options(op) #reset
149

```

		Df	Sum of Sq	RSS	AIC
- Transmission	1	0.023	35.176	4.917	
- Airbag	1	0.203	35.356	5.146	
- Owners	1	0.820	35.973	5.925	
<none>			35.154	6.888	
- KM	1	2.229	37.382	7.654	
- Fuel_type	2	10.281	45.435	14.433	
- Age	1	11.842	46.995	17.952	
- SR_Price	1	42.363	77.516	40.472	

Then we will have this 5.146, and then 5.9 to 5, and then the the existing model that is represented by none. So, none means if we do not drop any variable, then this is what we will have, right, all the variables are there. And so, we have 3 candidate models here, at trans when we drop transmission when we drop airbag and when we drop. So, rather so, when we transmission followed by this dropping of airbag and owners, these are the more these are the options alternatives that we have.

So, in the next one if you see that we have fuel type s R price k m plus owners plus airbag and age, we have 3 variables one has been dropped. And you can see the one that has been dropped is the transmission. And that was the first one, right, because we could achieve a lower a i c value 4.917. So, that particular one has been selected and transmission has been dropped.

Now, again from this model also. If we further drop airbag we would achieve a much lower a i c value of 3.147, we drop owners then we reach 23.93 and 9 and then if we do not drop anything then this is the model that we have at present. So, 2 candidate models seem to be performing better than the present model, with respect to the a i c value a i c criteria. Now if we again drop airbag to get the first model. So, you would see , that we reach the a i c value of 3.15, and the variables that we have in the model are fuel type, then s R price then k m then owners and then age. So, if we look at the options that we have is the owners if we drop owners.

Then we will have the a i c value of 2.113 which is less than the value for the current model 3.15. Then the current model is as you can see the 3.147 that is 3.15. Then further we can drop kilometer or we can add airbag. So, this is step we have already followed we will reach to the previous models. So, probably we will select the first one and we will drop the owners.

And you will see that reach this step a i c value of 2.11, and now the variables that we have is fuel type plus s R price plus k m plus age. So, these are the 4 variables is superior you remember. The model the final model that we got using reg subsets of function, that we had there also using adjusted R square criteria criterion the final model that we selected was of 4 variable model having the same variable, right, a s R price k m age and fuel type right.

So, in this case also as you can see the first the current model having 2.11, and there is no other model we can see the first row right among the options that we have the first row none that is the same model. So, no other model can improve this further. So, using a different criterion. So, we talked about the mallow c p adjusted R square and R square , using different criteria like a i c we also get the same results and in this case by running stepwise position.

Now as I said that in the in the results that we are getting here, they are they are with respect to the sample that we have very small sample we are dealing with very small samples and the small number of observation and also the partitioning. So, as I said if we change the partitioning, the number of the observation that are randomly selected in the training partition if they change the results might also change. If you want to see what will happen if we change the partitioning, we can repeat few of the models that we have just done. So, let us change the partitioning again, again if we do this partitioning again.

So, we have reek generated these partitions, also as you can see regeneration has happened. Now if we look at the model let us look at the let us go back to the same point variable selection if we look at the exhaustive search. So, once the partitioning has been done.

(Refer Slide Time: 25:22)

```

76 # Exhaustive search
77 library(leaps)
78 mod3sumsubsets(price ~., data = dtrain, nbest = 1, mmax = NULL,
79 force.in = NULL, force.out = NULL,
80 method = "exhaustive", intercept = T)
81 mod3sum=summary(mod3)
82
83 countspsch=function(x) sum(x=="A")
84 om=as.integer(apply(mod3sum$outmat, 2, countspsch)); om
85 data.frame("#Coeffs"=as.integer(apply(mod3sum$outmat, 1, countspsch)),
86 "RSS"=mod3sum$rss,
87 "Cp"=round(mod3sum$cp, digits=2),
88 "R-sq"=round(mod3sum$rsq, digits=2),
89 "Adj. R-sq"=round(mod3sum$adjr2, digits=2),
90 mod3sum$outmat[,order(-om)])
91

```

```

Console
C:\Users\user\Desktop\MOOC January 2018\Dr. Gaurav Dhill/Sessions 6\
> mod3sum
  <none>          1  1.673  37.796  2.150
+ Owners       1  0.767  35.356  3.147
+ Airbag       1  0.139  35.985  3.939
+ Transmission 1  0.002  36.122  4.111
- Fuel_type    2  10.346  46.470  9.446
- Age          1  11.553  47.677 12.601
- SR_Price     1  72.559 108.683 49.680
> partidx=sample(1:nrow(df), 0.6*nrow(df), replace = F)
> dtrain=df[partidx,]
> dtest=df[-partidx,]
>

```

We can read on this and you would see that, the results that we might get or might be slightly different. So, you will have to look at this.

(Refer Slide Time: 25:35)

```

79 force.in = NULL, force.out = NULL,
80 method = "exhaustive", intercept = T)
81 mod7sum=summary(mod7)
82
83 countspsch=function(x) sum(x=="A")
84 om=as.integer(apply(mod7sum$outmat, 2, countspsch)); om
85 data.frame("#Coeffs"=as.integer(apply(mod7sum$outmat, 1, countspsch)),
86 "RSS"=mod7sum$rss,
87 "Cp"=round(mod7sum$cp, digits=2),
88 "R-sq"=round(mod7sum$rsq, digits=2),
89 "Adj. R-sq"=round(mod7sum$adjr2, digits=2),
90 mod7sum$outmat[,order(-om)])
91
92 # Coefficients of subset models
93 # second argument is index vector indicating ordering in summary output
94 coef(mod7, 1:8)
95

```

```

Console
C:\Users\user\Desktop\MOOC January 2018\Dr. Gaurav Dhill/Sessions 6\
1  112.12885 17.72 0.49 0.48 *
2  90.04623 8.15 0.59 0.57 " "
3  79.44530 4.60 0.64 0.61 " " "
4  73.68502 3.58 0.66 0.63 " " "
5  70.90284 4.13 0.68 0.63 " " "
6  70.14625 5.73 0.68 0.63 " " "
7  69.22423 7.25 0.68 0.62 " " "
8  68.74909 9.00 0.69 0.62 " " "
Airbag Fuel_typePetrol Owners Transmission1
1  ( 1 )
2  ( 1 )

```

You can see how the numbers have completely changed. At least the numbers of these different criteria have changed you can see different partition you can see adjusted R square now I start with the 0.48, right. The earlier one was different right, we have the previous results as well and the output. So, if we go back you can find out t v s results,

yes, we started with 0.48, then 0.64 then 0.71, 0.64 than 0.71, and then 0.72 now there is just that we have now you can see that start with 0.48, then we these 2.57 than 0.61.

So, we do not reach to that 0.72 level, and you see that 4-variable model we reached the a peak value our gestures by 0.63. And then it remains 0.63 for 5 variable model and 6 variable model, especially if we look at only the 2 decimal point. And then you would see for 7 variable model and 8 variable model the value drops again.

So, we will have to select you know 4 variable model to 6 variable models we had 3 options, in this case you can see just by changing the partitioning, instead of you know using the adjusted R square as the criterion. Now we have to pick from these 3 model instead of just one in the previous scenario previous partitioning that, we did we look at the R square value that is also. There also the results in is remains to be the same that 5 variable model having 0.68 value is going to be selected if we look at the c p value.

So, those numbers have changed significantly. Now we look at c p value 3 variable model the value is 4 points I 6, and then the this value we need to compare is 4, and difference is point 6, right. And we look at the next value. So, it will be compared with 5 and the value is this one 3 point 5 8. So, therefore, difference of for more than 1.

So, probability variable model is again going to be selected here in this case as well, but if we look at the variables now, the column for sr price you would see it is still present. So, it still present and then age is also present and followed by km. Now if you see that km is present in 5 of the models and you see the 5 fuel type of petrol, let us present in the 4 and 4 in the models, if you go back to the previous results that we have had, it was the fuel type of petrol which represent in the 5 you know 6 of the models, and k m was in the 5 of the models right 5 of the 8 models. So, that has changed.

So, fuel the s R price and is they are still present in 8 models and say and models respectively. But k m and fuel type of petrol they have you know they have changed their places right, who k m coming into 5 models and a fuel type of petrol in 4 models. So, you can see that once we change the partitioning the results changed. And this is mainly due to the small sample size that we are dealing with if we had a much larger size probably it would not change the spite partitioning, because we have we would have more number of observations to learn from to build our model from. And therefore, the results are going to be more robust right.

So, with this the same thing you can apply on other algorithm that we have discussed, and that with the change in the partitioning the results would also change. So, with this we will like to stop here.

So, this also concludes our discussion on multiple linear regression all right. So, in the next session will start with  $k \ll n$ .

Thank you.