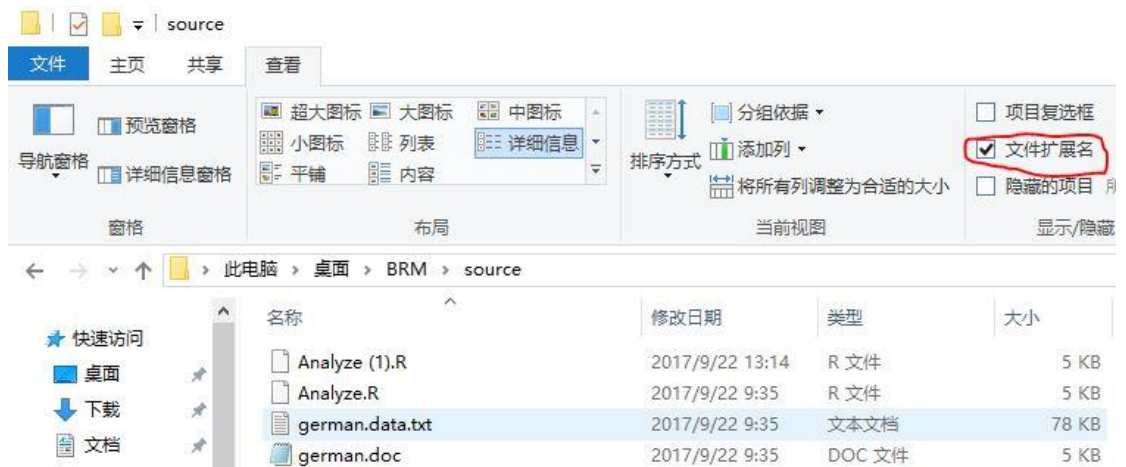


## 题目：商科研究方法研究专题 2

### R-studio 操作

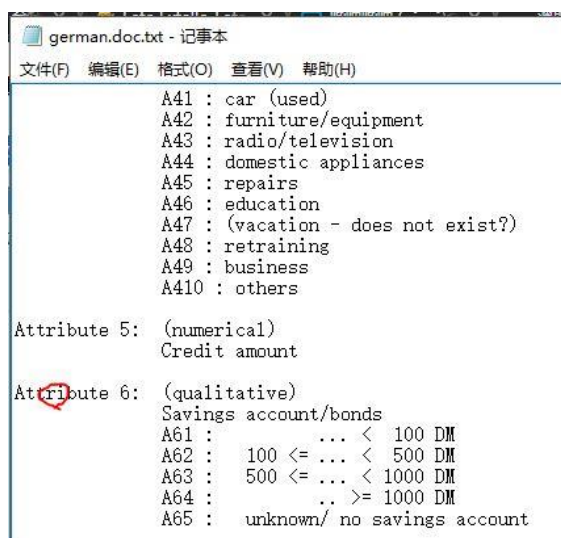
- 一， 安装 R, R-studio, Wake(下载地址看 PPT)。不同操作系统的操作界面会有所不同，按具体操作选项执行。
- 二， 下载执行用原始数据，Analyze.R, readcolnames.R, german.data, german.doc。
- 三， 在打开 R-studio 前请保证关掉 360 等软件。
- 四， Analyze.R 中可能会出现执行指令的标记行数不符，可参照截图内容判断。

五， 检查拓展名。请注意：



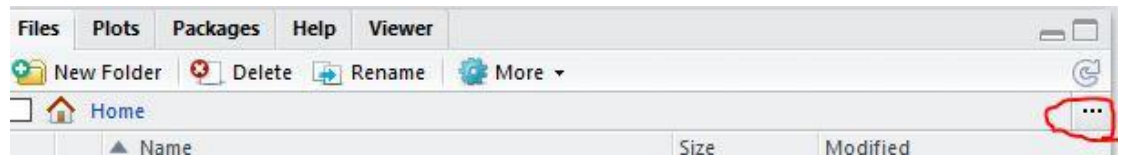
确认此状态下，german.doc 修改为 german.doc.txt。并且修改

数据中的缺少一个 r 的错误。如图：

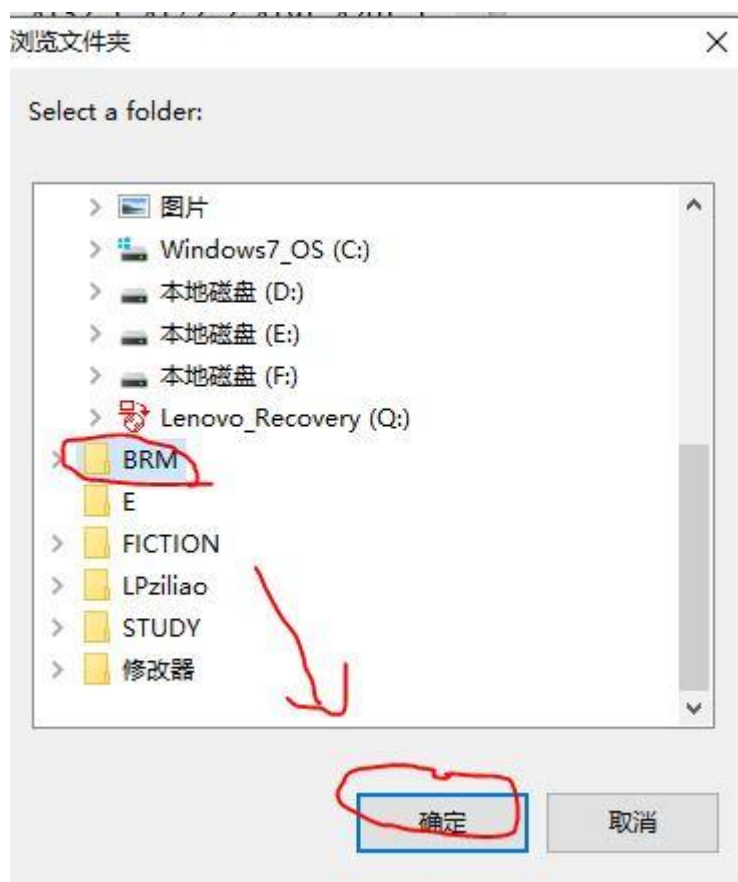


六， 打开 R-studio 以后， 在此处选择工作文件夹（操作系统不同具体位置不同）：

1. 点击

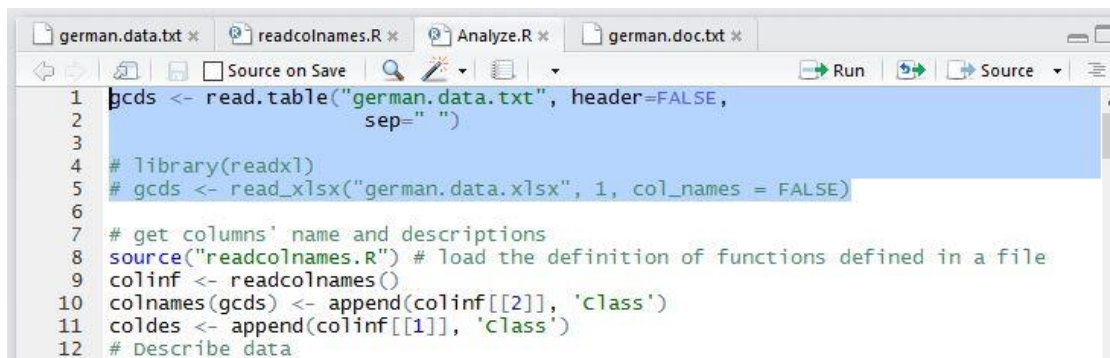


2. 确定选择工作文件夹：



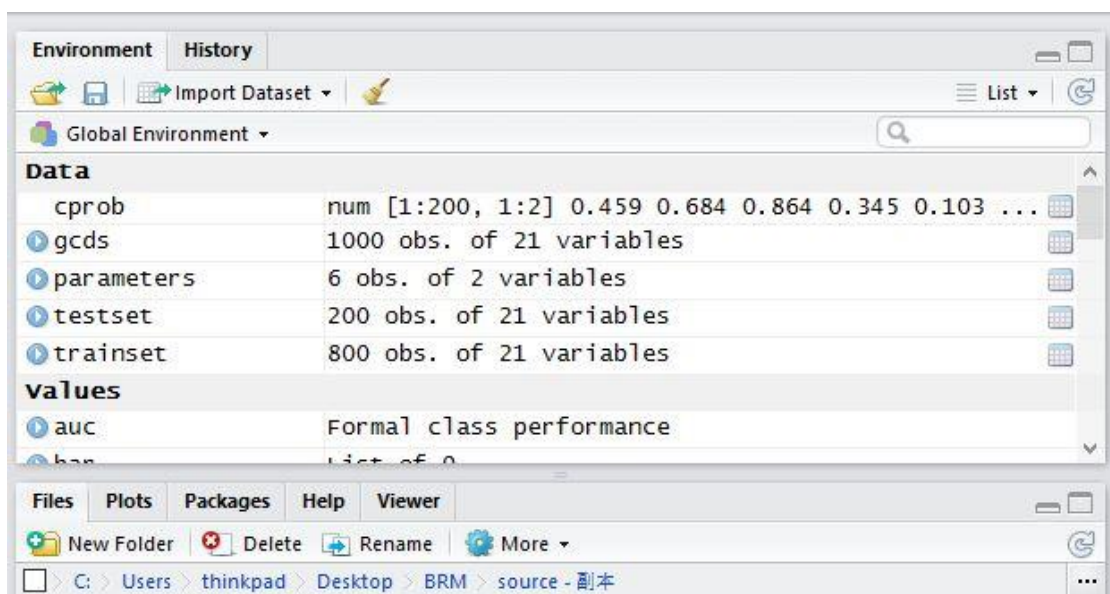
### 3. 载入原始数据

在数据查看版面点选 Analyze.R,如图选中 1-5 行, 点击 RUN。



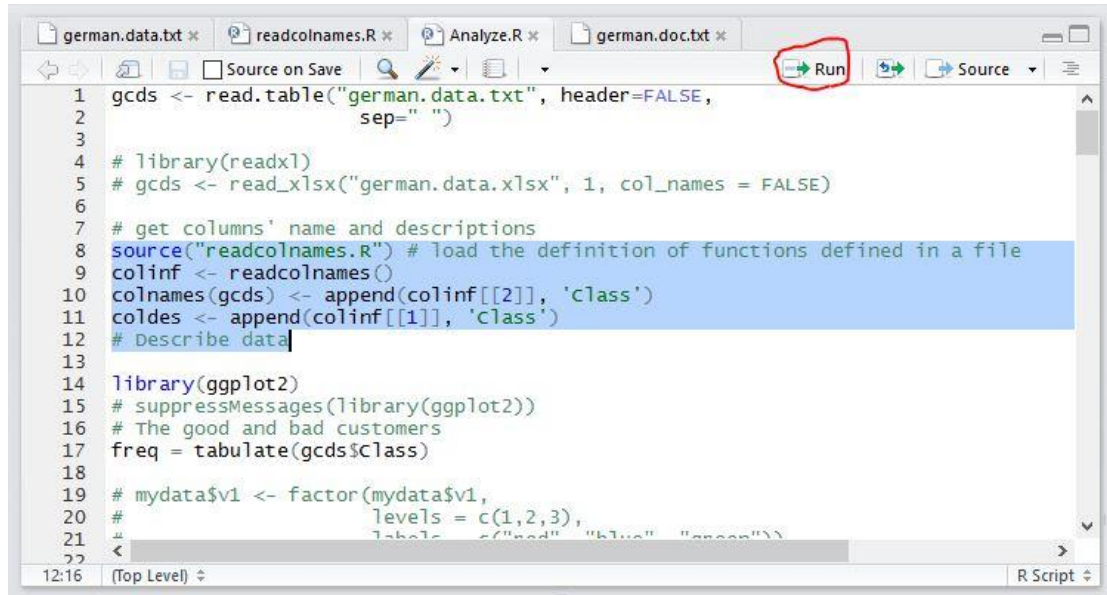
```
1 gcds <- read.table("german.data.txt", header=FALSE,
2                     sep=" ")
3
4 # library(readxl)
5 # gcds <- read_xlsx("german.data.xlsx", 1, col_names = FALSE)
6
7 # get columns' name and descriptions
8 source("readcolnames.R") # load the definition of functions defined in a file
9 colinf <- readcolnames()
10 colnames(gcds) <- append(colinf[[2]], 'class')
11 coldes <- append(colinf[[1]], 'class')
12 # Describe data
```

出现下图时, 证明数据已经载入:



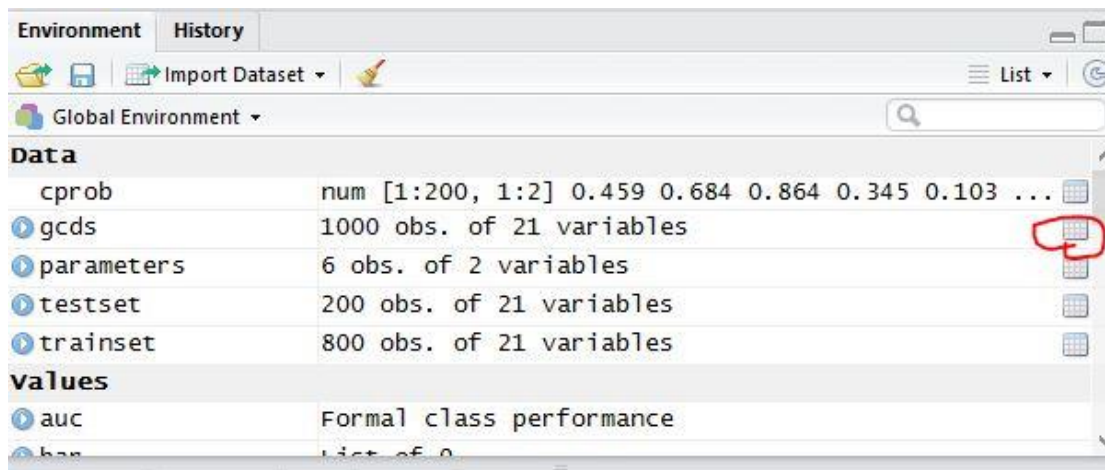
#### 4. 缩略变量名称

在数据查看版面点选 Analyze.R, 如图选中 8-12 行, 点击 RUN。



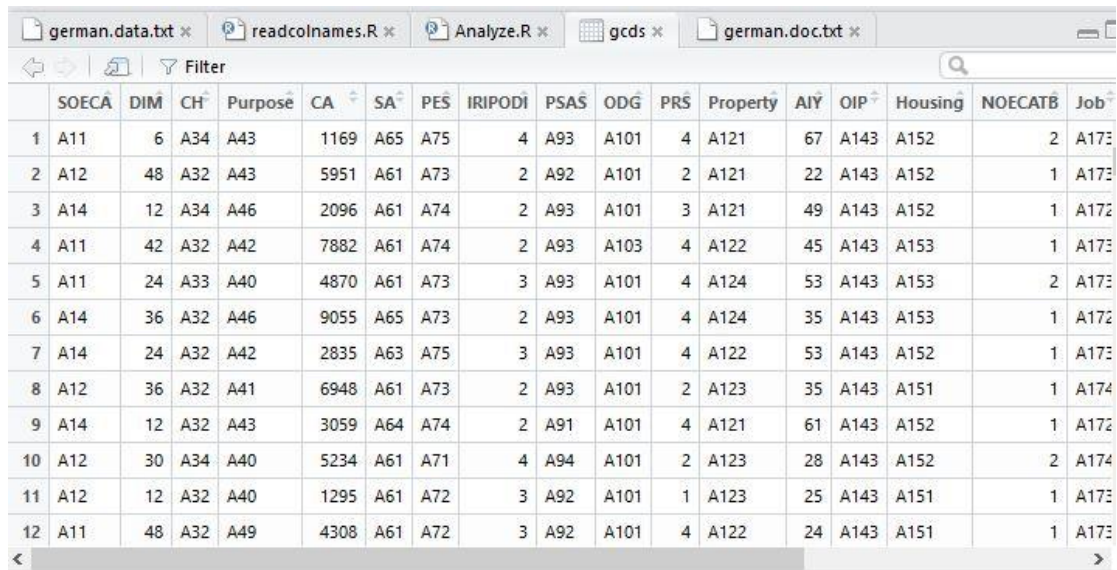
```
1 gcds <- read.table("german.data.txt", header=FALSE,
2                     sep=" ")
3
4 # library(readxl)
5 # gcds <- read_xlsx("german.data.xlsx", 1, col_names = FALSE)
6
7 # get columns' name and descriptions
8 source("readcolnames.R") # load the definition of functions defined in a file
9 colinf <- readcolnames()
10 colnames(gcds) <- append(colinf[[2]], 'class')
11 coldes <- append(colinf[[1]], 'class')
12 # Describe data
13
14 library(ggplot2)
15 # suppressMessages(library(ggplot2))
16 # The good and bad customers
17 freq = tabulate(gcds$class)
18
19 # mydata$y1 <- factor(mydata$y1,
20                     levels = c(1,2,3),
21                     labels = c("good", "bad", "neutral"))
22
23
```

打开 gcds 查看窗口



Environment		History	
Global Environment			
<b>Data</b>			
cprob	num [1:200, 1:2]	0.459 0.684 0.864 0.345 0.103 ...	
gcds	1000 obs. of 21 variables		
parameters	6 obs. of 2 variables		
testset	200 obs. of 21 variables		
trainset	800 obs. of 21 variables		
<b>values</b>			
auc	Formal class performance		
bar	List of 0		

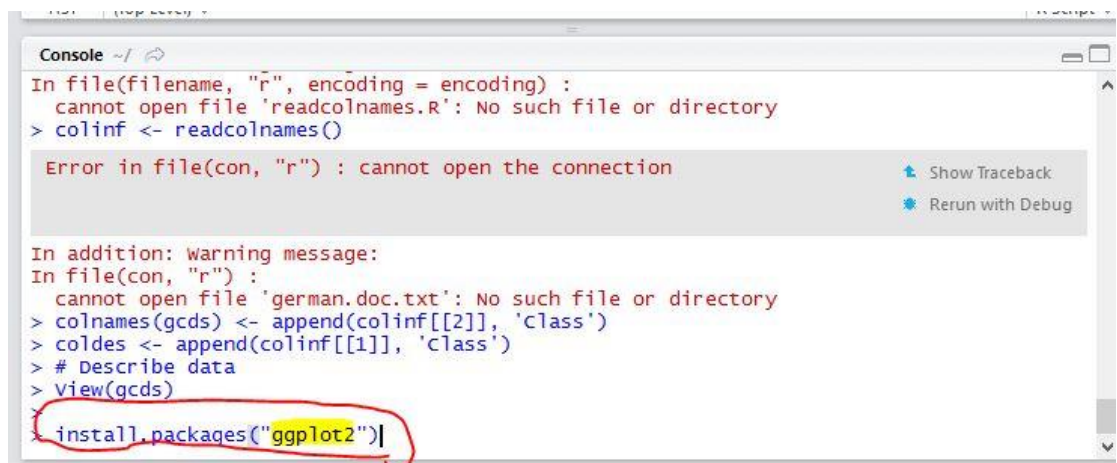
如图出现缩略变量名称如时，即成：



	SOECA	DIM	CH	Purpose	CA	SA	PE	IRIPODI	PSAS	ODG	PRS	Property	AIY	OIP	Housing	NOECATB	Job
1	A11	6	A34	A43	1169	A65	A75	4	A93	A101	4	A121	67	A143	A152	2	A173
2	A12	48	A32	A43	5951	A61	A73	2	A92	A101	2	A121	22	A143	A152	1	A173
3	A14	12	A34	A46	2096	A61	A74	2	A93	A101	3	A121	49	A143	A152	1	A172
4	A11	42	A32	A42	7882	A61	A74	2	A93	A103	4	A122	45	A143	A153	1	A173
5	A11	24	A33	A40	4870	A61	A73	3	A93	A101	4	A124	53	A143	A153	2	A173
6	A14	36	A32	A46	9055	A65	A73	2	A93	A101	4	A124	35	A143	A153	1	A172
7	A14	24	A32	A42	2835	A63	A75	3	A93	A101	4	A122	53	A143	A152	1	A173
8	A12	36	A32	A41	6948	A61	A73	2	A93	A101	2	A123	35	A143	A151	1	A174
9	A14	12	A32	A43	3059	A64	A74	2	A91	A101	4	A121	61	A143	A152	1	A172
10	A12	30	A34	A40	5234	A61	A71	4	A94	A101	2	A123	28	A143	A152	2	A174
11	A12	12	A32	A40	1295	A61	A72	3	A92	A101	1	A123	25	A143	A151	1	A173
12	A11	48	A32	A49	4308	A61	A72	3	A92	A101	4	A122	24	A143	A151	1	A173

5.从此开始，可能会缺少安装包，按指令安装即可。

例如 ggplot2，双引号内输入 ggplot2 即可



```
In file(filename, "r", encoding = encoding) :
  cannot open file 'readcolnames.R': No such file or directory
> colinf <- readcolnames()

Error in file(con, "r") : cannot open the connection
Show Traceback
Rerun with Debug

In addition: warning message:
In file(con, "r") :
  cannot open file 'german.doc.txt': No such file or directory
> colnames(gcds) <- append(colinf[[2]], 'class')
> coldes <- append(colinf[[1]], 'class')
> # Describe data
> view(gcds)
> install.packages("ggplot2")
```

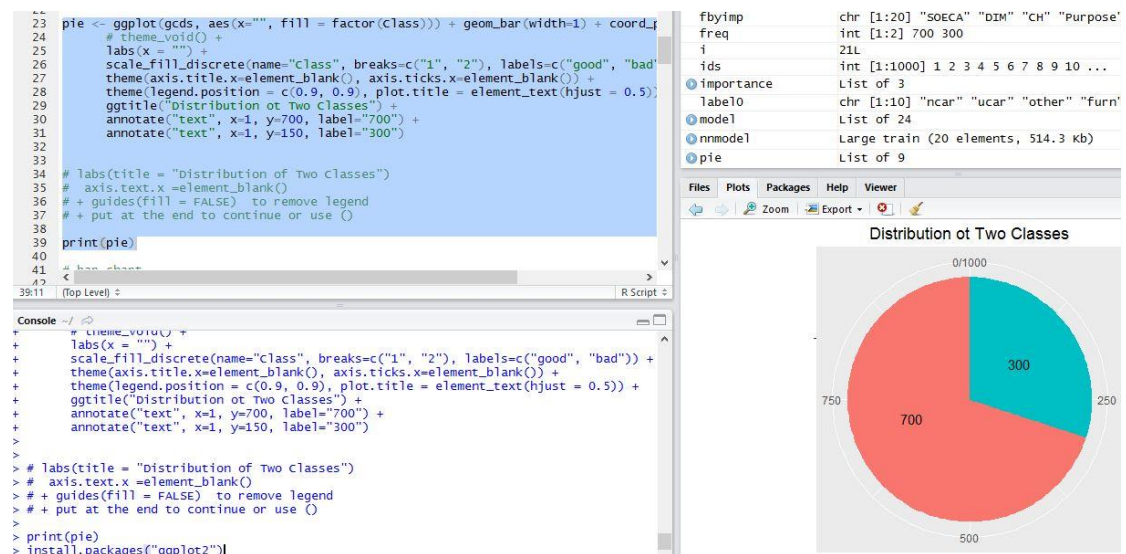
同理，课堂上已知需要安装的有 stringr, ggplot2, caret, LibinearR, e1071, tree, ROCR, 区分大小写，按系统要求安装即可。



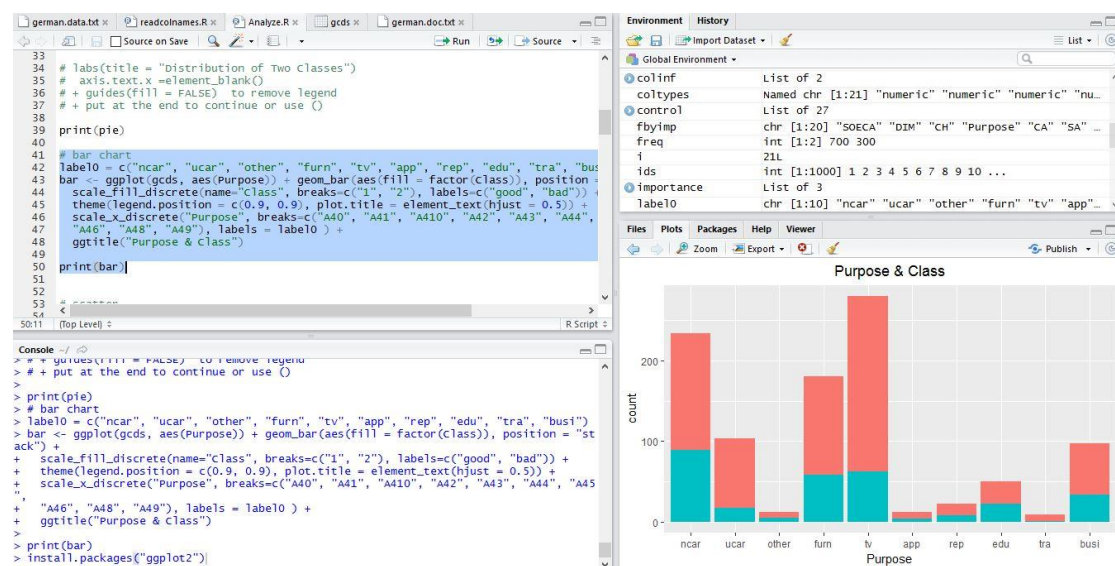
6.运行 ggplot2, RUN:14-21 行。

```
13 library(ggplot2)
14 # suppressMessages(library(ggplot2))
15 # The good and bad customers
16 freq = tabulate(gcds$class)
17
18 # mydata$y1 <- factor(mydata$y1,
19 #                     levels = c(1,2,3),
20 #                     labels = c("red", "blue", "green"))
21
22
23 pie <- ggplot(gcds, aes(x="", fill = factor(Class))) + geom_bar(width=1) + coord_l
24 # theme_void() +
```

7.画饼图, RUN: 23-39 行。



如果, Plots 窗口中出现 700 : 300 的饼图。同理条形图, RUN : 41-50 行。



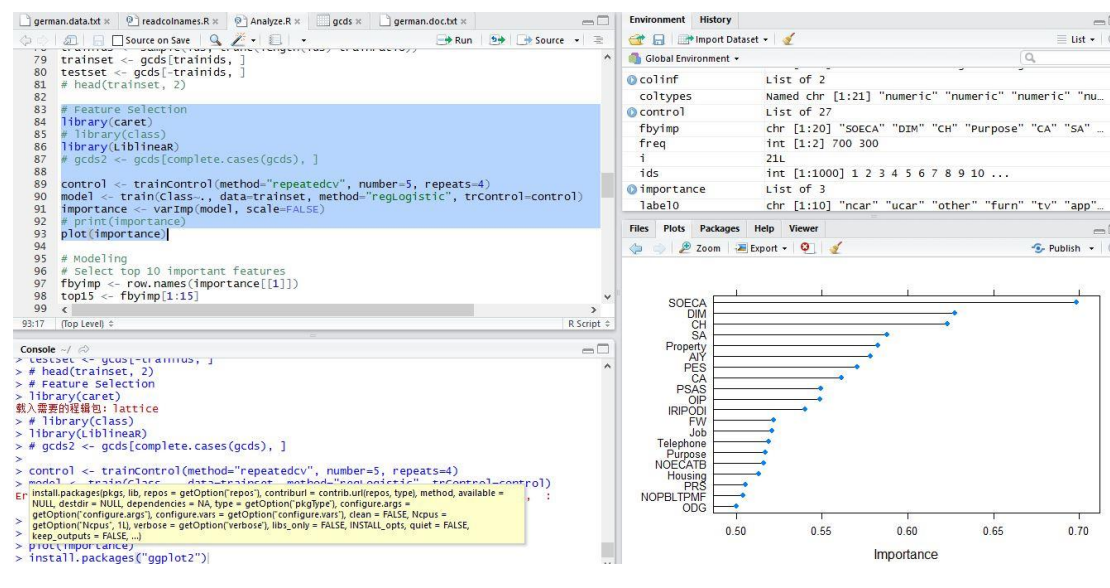
8.随机抽样， RUN：60-72 行。

```
60 # Sampling
61 # tranform character column to factor
62 coltypes <- sapply(gcds, mode)
63 for (i in 1:ncol(gcds))
64 {
65   if (coltypes[[i]] == "character")
66     gcds[[i]] <- factor(gcds[[i]])
67 }
68
69
70 # gcds$class[gcds$class == 1] <- 'Good'
71 # gcds$class[gcds$class == 2] <- 'Bad'
72 gcds$class <- factor(gcds$class, levels=c(1,2), labels=c("Good", "Bad"))
73
74
```

9.区分测试样本和训练样本， 指令中是分为 80%训练样本， 20%测试样本。 RUN：75-81 行。

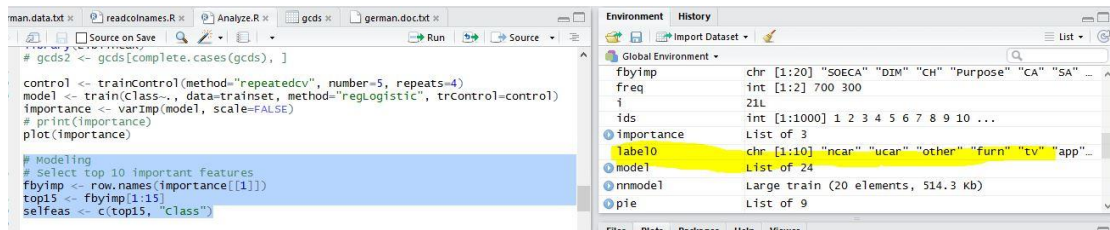
```
73
74
75 set.seed(2)
76 ids <- 1:nrow(gcds)
77 trainratio <- 0.8
78 trainids <- sample(ids, trunc(length(ids)*trainratio))
79 trainset <- gcds[trainids, ]
80 testset <- gcds[-trainids, ]
81 # head(trainset, 2)
82
83 # Feature Selection
```

10.特征选择， 如图右下输出 importance 统计表。 RUN：83-93 行。



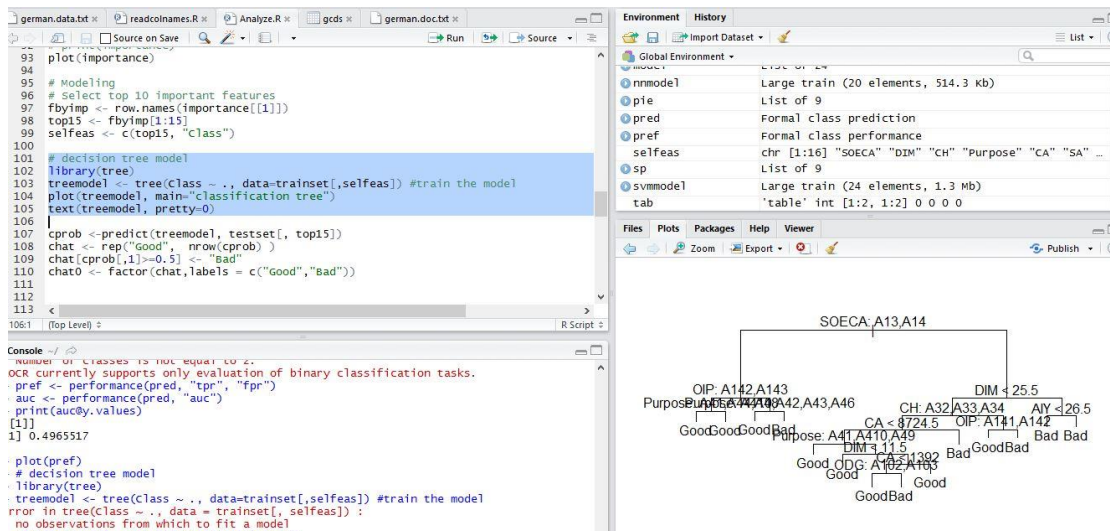


11.挑选十个最重要的变量，RUN：95-99 行。



如上图所示，挑选的十个最重要变量可在右边窗口的黄色高光栏中查看。

12.决策树，RUN：101-105 行，决策树输出如右下：



```
cprob <- predict(treemodel, testset[, top1$])
chat <- rep("Good", nrow(cprob))
chat[cprob[,1]>=0.5] <- "Bad"
chat0 <- factor(chat, labels = c("Good", "Bad"))
```

```
library(ROCR)
```

```

109 chat[cprob[,1]>=0.5]] <- "Bad"
110 chat0 <- factor(chat,labels = c("Good","Bad"))
111
112
113 library(ROCR)
114 tab <- table(chat0, testset$class)
115 print(tab)
116
117 pred <- prediction(as.numeric(chat0), as.numeric(testset$class))
118 pref <- performance(pred, "tpr", "fpr")
119 auc <- performance(pred, "auc")
120 print(auc@y.values)
121 plot(pref)
122
123
124 # support vector machine
125
126 control <- trainControl(method="cv", savePredictions = T, ClassProbs = T)
127 svmmodel <- train(Class~., data=trainset[, selfseas], method="svmRadialsigma",
128 <
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

右下图示的曲线划分出的图中面积（右下） $>0.5$ ，面积越接近 1，解释度越好。

15.支持向量机模型判断，如图高光所示，解释度为 0.72.

```
123
124 # support vector machine
125
126 control <- trainControl(method="cv", savePredictions = T, classProbs = T)
127 svmmodel <- train(Class~., data=trainset[, selfeas], method="svmRadialSigma",
128                   trControl=control)
129
130 chat <- predict(svmmodel, newdata = testset[, top15])
131
132
133 tab <- table(chat, testset$class)
134 print(tab)
135
136 pred <- prediction(as.numeric(chat), as.numeric(testset$class))
137 pref <- performance(pred, "tpr", "fpr")
138 auc <- performance(pred, "auc")
139 print(auc@y.values)
140 plot(pref)
141
142
```

140:11 (Top Level) R Script

Console ~/  
> tab <- table(chat, testset\$class)  
> print(tab)

chat	Good	Bad
Good	144	55
Bad	1	0

>  
> pred <- prediction(as.numeric(chat), as.numeric(testset\$class))  
> pref <- performance(pred, "tpr", "fpr")  
> auc <- performance(pred, "auc")  
> print(auc@y.values)  
[[1]]  
[1] 0.4965517  
  
> plot(pref)  
> install.packages("kernlab")

16, NNmodel, 神经网络。得到输出图，如输出数据所示，神经网络模型解释度为 0.795。

```
143 # neural networks model
144 library(caret)
145 parameters<- expand.grid(.decay = c(0.5, 0.1), .size = c(5, 6, 7))
146 nnmodel <- train(trainset[, top15], trainset[, "class"],
147                 method="nnet", tuneLength = 10, trace = F, maxit = 1000,
148                 tuneGrid= parameters)
149 chat <- predict(nnmodel, newdata = testset[, top15])
150
151
152 tab <- table(chat, testset$class)
153 print(tab)
154
155 pred <- prediction(as.numeric(chat), as.numeric(testset$class))
156 pref <- performance(pred, "tpr", "fpr")
157 auc <- performance(pred, "auc")
158 print(auc@y.values)
159 plot(pref)
160
161
162
```

159:11 (Top Level) R Script

Console ~/ ↵

```
>
> tab <- table(chat, testset$class)
> print(tab)
chat    Good Bad
Good   126  22
Bad     19  33
>
> pred <- prediction(as.numeric(chat), as.numeric(testset$class))
> pref <- performance(pred, "tpr", "fpr")
> auc <- performance(pred, "auc")
> print(auc@y.values)
[[1]]
[1] 0.7344828
> plot(pref)
> install.packages("kernlab")
```