CDA 532 Homework 1

UB ID: 50413349

Name: Shreeram G S

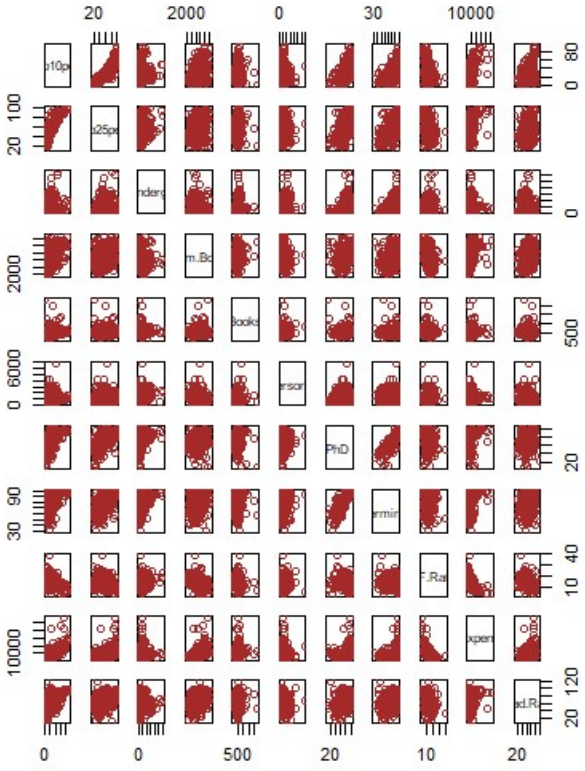
Q1) First we shall load all essential packages such as ISLR which holds the College data set, and caret, tidyverse, ggplot (for histograms). To load the dataset of college, we use the data() function. The head() function below gives us from 0 - 5 tuples of our dataset. We split the data set into 2 parts for our pairwise plots further, where we compare the two segments.

```
> require(ISLR)
> require(caret)
> require(tidyverse)
> library(ggplot2)
> data(College)
> head(College)
                             Private Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD Terminal S.F.Ratio perc.alumni
Abilene Christian University     Yes 1660   1232    721        23        52        2885         537     7440       3300   450     2200  70       78      18.1          12
Adelphi University               Yes 2186   1924    512        16        29        2683        1227    12280       6450   750     1500  29       30      12.2          16
Adrian College                   Yes 1428   1097    336        22        50        1036          99    11250       3750   400     1165  53       66      12.9          30
Agnes Scott College              Yes  417    349    137        60        89         510          63    12960       5450   450      875  92       97       7.7          37
Alaska Pacific University        Yes  193    146     55        16        44         249         869     7560       4120   800     1500  76       72      11.9           2
Albertson College                Yes  587    479    158        38        62         678          41    13500       3335   500      675  67       73       9.4          11
                             Expend Grad.Rate
Abilene Christian University   7041        60
Adelphi University            10527        56
Adrian College                 8735        54
Agnes Scott College           19016        59
Alaska Pacific University     10922        15
Albertson College              9727        55
> set.seed(1)
> var <- createDataPartition(College$Accept , p = 0.5, list = FALSE)
> |
```

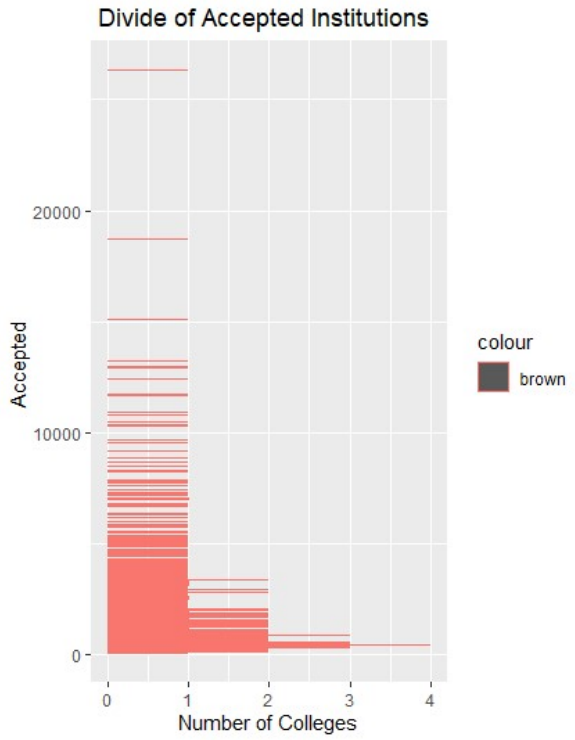
a) After splitting we shall use the pairs() function to perform pairwise plots and use ggplot() to make histograms. Here we have taken Accept column and phD column from the college data set to create histograms.

```
> #Q1.a)
> #pairwise plots
> pairs(College[var,-var],col="brown")
> #histograms: Accepted vs number of college
> var1 <- ggplot(College, aes(y=Accept, col="brown" ))+
+   geom_bar()+
+   ggtitle(" Divide of Accepted Institutions")+
+   labs(x="Number of Colleges",
+        y="Accepted ")
> var1
>
> #histograms of colleges phd candidates
> var1 <- ggplot(College, aes(x=PhD))+
+   geom_bar()+
+   ggtitle("Count of Institutions per phd candidates")+
+   labs(x="PhD",
+        y=" Count of Colleges")
> var1
```

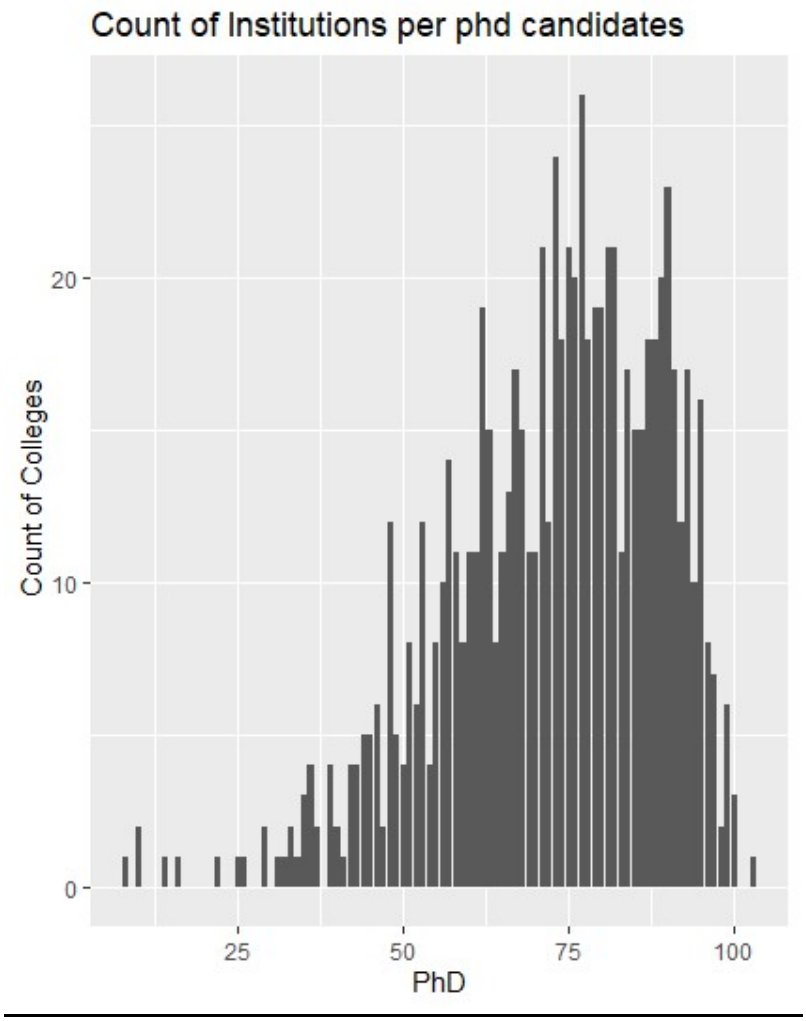
*__PAIRWISE PLOTS__:



***Histogram of Accepted vs number of college:**



Histograms of colleges PhD candidates:

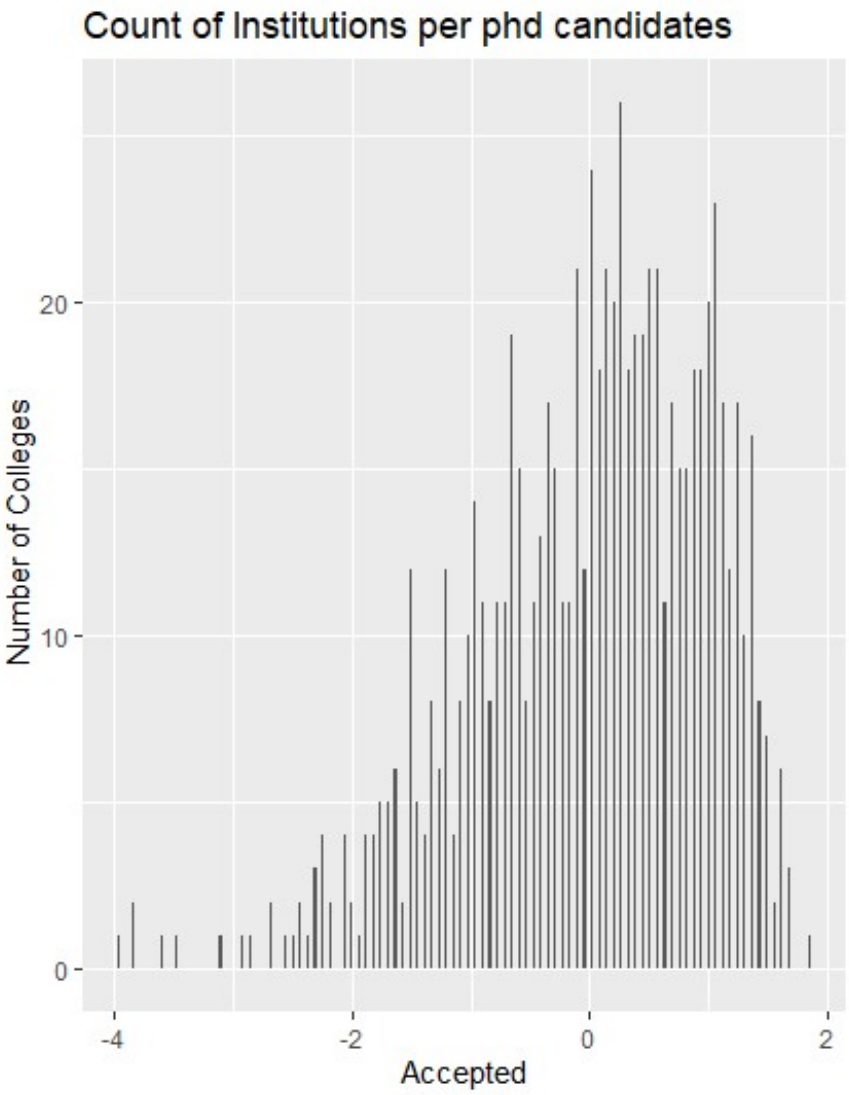


Yes, clearly even after viewing both the columns of the dataset, we see it is difficult to come to conclusions as the graph has too much noise. It would be cleaner to perform scaling to get an evenly distributed graph, which would make it easier to read and understand hence consequently allowing to make decisions based on it.

b) Scaling Accepted number of the colleges and plotting histogram:

```
> #Scaling PhD candidates of the colleges and plotting histogram
> var1<-scale(College$PhD, center = TRUE, scale = TRUE)
> var2 <- ggplot(College, aes(x=var1))+
+   geom_bar()+
+   ggtitle("Count of Institutions per phd candidates")+
+   labs(x="Accepted",
+        y=" Number of Colleges")
> var2
```

Now after scaling we see each of the accepted applications are more sparsely seen, and we can differentiate and make much more accurate decisions. Now we see the range between [-4,2] and making it simpler to read the graph.



c) Here we are splitting the private column of yes/no into private and public and converting the main data set into 2 parts. We use the save() function to save it as .Rdata in the file = " " part inside the save function. Now these data set can further separately be used to be loaded or to perform further operations on it.

```
> #1.c)
> head(College$Private)
[1] Yes Yes Yes Yes Yes Yes
Levels: No Yes
> #Creating Private Data frame
> private<-College[College$Private == 'Yes',]
> head(private)
                          Private Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD Te
Abilene Christian University     Yes 1660   1232    721        23        52        2885         537     7440       3300   450     2200  70
Adelphi University               Yes 2186   1924    512        16        29        2683        1227    12280       6450   750     1500  29
Adrian College                   Yes 1428   1097    336        22        50        1036          99    11250       3750   400     1165  53
Agnes Scott College              Yes  417    349    137        60        89         510          63    12960       5450   450      875  92
Alaska Pacific University        Yes  193    146     55        16        44         249         869     7560       4120   800     1500  76
Albertson College                Yes  587    479    158        38        62         678          41    13500       3335   500      675  67
                          Expend Grad.Rate
Abilene Christian University   7041        60
Adelphi University            10527        56
Adrian College                 8735        54
Agnes Scott College           19016        59
Alaska Pacific University     10922        15
Albertson College              9727        55
> save(private, file = "private.Rdata")
> #Creating Public Data frame
> public<-College[College$Private == 'No',]
> head(public)
                                Private  Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books Perso
Angelo State University              No  3540   2001   1016        24        54        4190        1512     5130       3592   500     2(
Appalachian State University         No  7313   4664   1910        20        63        9940        1035     6806       2540    96     2(
Arizona State University Main campus No 12809  10308   3761        24        49       22593        7585     7434       4850   700     2(
Arkansas Tech University             No  1734   1729    951        12        52        3602         939     3460       2650   450     1(
Auburn University-Main Campus        No  7548   6791   3070        25        57       16262        1716     6300       3933   600     1(
Bemidji State University             No  1208    877    546        12        36        3796         824     4425       2700   660     1(
                                perc.alumni Expend Grad.Rate
Angelo State University                   5   4010        34
Appalachian State University             14   5854        70
Arizona State University Main campus      5   4602        48
Arkansas Tech University                  5   4739        48
Auburn University-Main Campus            18   6642        69
Bemidji State University                 16   3752        46
> save(public, file = 'public.Rdata')
```

d) Here we use the order() function and set decreasing = TRUE to make the descending order of both the dataset with respect to column apps. The method function is used to select which sorting technique to be implied. There are mainly 3 options: “Auto”, “shell” and “radix”.

```
#1.d) ordering private in descending order
order(private$Apps, decreasing = TRUE, method = c("auto"))
#ordering in public in decreasing order
order(public$Apps, decreasing = TRUE, method = c("shell"))
```

Below we see the result after running the above code.

```
> #1.d) ordering private in descending order
> order(private$Apps, decreasing = TRUE, method = c("auto"))
  [1]  53 482 214 153 336 373  63 494 345 502 342 188 564 457 498  78 139 166 242 358 186 513 515 493 529 221  62 238 209 490 473 125  65 299 262 481 479  35 497 522 178 290
 [43] 374 487 382 237 107 169 348 537 289 422 501 459  17 433 430 291 175  57 555 462 274 496 415 254 499  82  85 362 275  56 439 377 319 454 163 190 384 277 514 477  43  54
 [85] 527 390 403 122 164 206 469 233 130 144 120 565 483 133 425 281 351 253 431 533 101  74 106 150 266 561 142 444 316  77 220  12 446 332 401  33 202 182 506 285 472 241
[127] 160 140  67 349 357 393 500 383 303 475 408 387 338 247 352   2 103 181 151 550  23 421 121 563 455 486 395  72 213 512 503 201 273 145 143 306 559 380 562 172  32 407
[169]   8 276 192  25 547 100  44 516 322  71 391 378 460 453 495 232 280  84  75 346 307 545  11 212 224 508 204  30 548 414   1 552 540  48 272 551  81 288 409 359 489 134
[211] 538 270 304 452 278 132 364 560 355 441 488 535 260 517  64 519 296 161 249 507 310  59 108 544   3 159  16 521 171 510 261 173 416 215 443 154 170 448 327  40 314 330
[253]  87 526 279  91  14 468 437 259 447 557 318 525  39 312  18 492 528 539 203  99 530 420  13 328  36 235  79 267 465 195 368 371  19 340 549 183 333 417 298  80 434 523
[295]  83 372 397 251   9 376 428  93  69 149 104 207 210 244 284 427 300 542 135  34 116 536 256 263 505 286  94 225  27 335 110 361 295 360 113 423 388 398 484 219 218 399
[337] 543 356 311 546 271 394 179 231 404 321  98  45 405 370 283 337 136 450  66 344 305 419  73 466  61 363 400 258 287 217 208 216 339 269 185 369  37  49 177 520 109 146
[379] 350 268 411 198  86 354 165  26 467 353 282 124 236 187 229  20  22 152 410 223 379 294  52 365 128 478 480 193 119 184 532 366 509 553  24 541 167 329 449 518 191  38
[421]  41  28 474  21 343 309 199 226 230 240 105 148  29 222  90 315 228  68 313 464 245   6 385 320 138 246 418  10 301 485 141  70 168 389 197 176 297 196 117 255 424 129
[463]  42 158 375 334 114 200 331 137  47  50 292 257 463  89  15 412 157 386 476 293 265 239 554 211 341 426 456 248 102 156 112 491 227 451 115  55 131 194 524 429 392 381
[505] 302   4 367  60 432 205 155 347  51 504 440  88 325  58 252 147 250   7 189 180 111  96 413 308 326 511 442 435 534 123 243 402 174 438 126 531 323 234 317  31 458 470
[547] 162 264 324 127 461 471 396 436  95  46   5 445  92 556 558 406 118  76  97
> #ordering in public in decreasing order
> order(public$Apps, decreasing = TRUE, method = c("shell"))
  [1]  92  88 129  85 148  60  40 186 199 130 109 137 193 182 160 121 108 146 145 107   3 187 110 134  31  42 135 151  82  79  75 158 133 172  20  97  23   8  59 205  38 170
 [43] 116 171 139  41 117 138  16 112 114  35   9 175 185 176   5 180   2 111  19 131 132   7 153 200 202 157 103 163  77  62  52  78  45  98 161  94  89  39  34  26 118 207
 [85] 167 113 194  68 162  37 184 125 197 115  74 126 140  18 169  13  10  83 166 191  56 123 144  63 183  96 149  12 168 106  95 152  91  36 136  50   1  67  32  93 147  24
[127] 210  30 181 128 204 119  46 208  55  81 198  76  43 179  99  14 164  51  80 206  61 189 102  69 192  87  53 178  47 211 100  70 212  25 120  90 195  71 154  28  33 165
[169] 127 159   4  49 155  58  65  73 188 124 150  29 177 201 156 105  11  27   6 141  48 203 104  21 209  84 190  15 174  44 101  72  86  17  66  54 122  64 143 142  22 196
[211] 173  57
```

e) Now we eliminate the rest of the records which are not of median of Top25perc column of the college data set. Then we shall store it in a new variable called “private/public”

```
#1.e)Eliminate all colleges of median less than Top25%
private_elimnated<-College[median(private$Top25perc),]
summary(private)
public_elimnated<-College[median(public$Top25perc),]
summary(public)
```

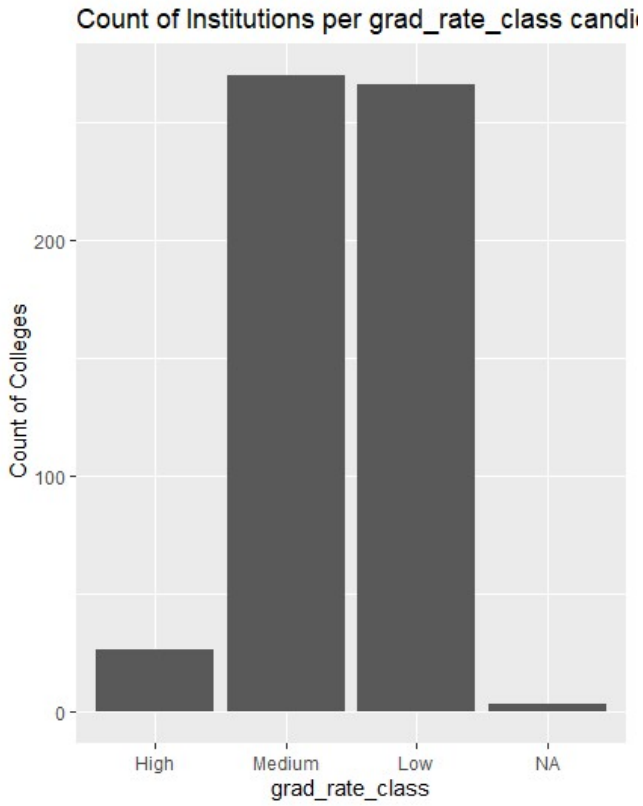
f) Here we use the ordered to order function after the data is been cut. The cut() function allows us to cut data into bins and specify 'cut labels'. There should n+1 bins if there are n labels. So here as you can see, the bins are of the size 15,40,70,100 and the labels are high low and medium. Further we have append it to public and private datasets.

Later we plot histogram and we see that now we can make even easier and simpler decisions as there are only 3 labels.

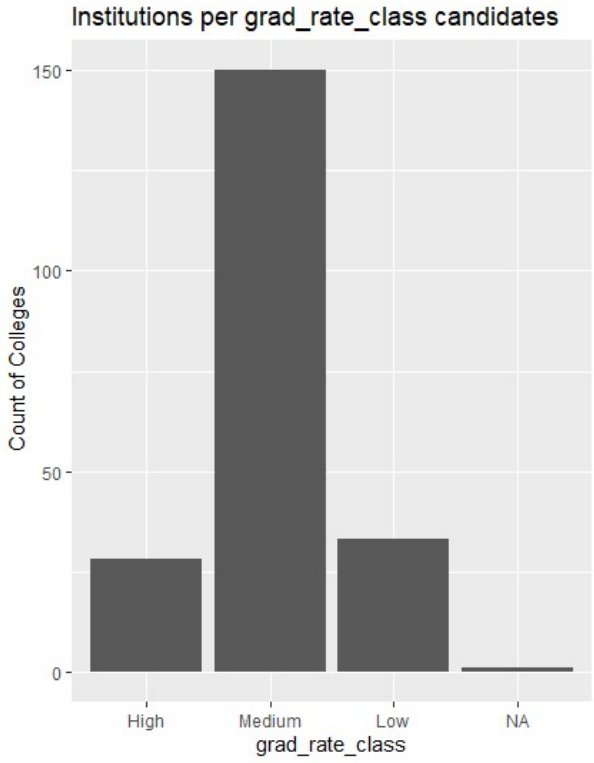
```
#1.f)graduation rate into high or low
#grad_rate_class of private
grad_rate_class <- ordered(cut(private$Grad.Rate, c(15,40, 70, 100)), labels = c("High", "Medium", "Low") )
summary(grad_rate_class)
#append to private
private$grad_rate_class <-grad_rate_class
head(private)
var1 <- ggplot(private, aes(x=grad_rate_class))+
  geom_bar()+
  ggtitle("Count of Institutions per grad_rate_class candidates")+
  labs(x="grad_rate_class",
       y=" Count of Colleges")
var1
#grad_rate_class of public
grad_rate_class <- ordered(cut(public$Grad.Rate, c(15,40, 70, 100)), labels = c("High", "Medium", "Low") )
summary(grad_rate_class)
#append to public
public$grad_rate_class <-grad_rate_class
head(public)
var1 <- ggplot(public, aes(x=grad_rate_class))+
  geom_bar()+
  ggtitle("Count of Institutions per grad_rate_class candidates")+
  labs(x="grad_rate_class",
       y=" Count of Colleges")
var1
```

Below there are plots of private and public data sets after being appended, respectively.

Private:


Count of Institutions per grad_rate_class candi
200
100
0
Count of Colleges
High
Medium
Low
NA
grad_rate_class

Public:


Institutions per grad_rate_class candidates
150
100
50
0
Count of Colleges
High
Medium
Low
NA
grad_rate_class

g) First we create a new list "new_list" using the list() function. Then we shall append the data frames private and public respectively as shown below.

```
> #create a list
> new_list<-list()
> #append private data set to list
> new_list[[length(new_list)+1]] <- private
> #append public data set to list
> new_list[[length(new_list)+1]] <- public
> new_list
```

Below we can see the result of performing this operation.

```
                                            Private Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD Terminal S.F.Ratio
Abilene Christian University                    Yes 1660   1232    721        23        52        2885         537     7440       3300   450     2200  70       78      18.1
Adelphi University                              Yes 2186   1924    512        16        29        2683        1227    12280       6450   750     1500  29       30      12.2
Adrian College                                  Yes 1428   1097    336        22        50        1036          99    11250       3750   400     1165  53       66      12.9
Agnes Scott College                             Yes  417    349    137        60        89         510          63    12960       5450   450      875  92       97       7.7
Alaska Pacific University                       Yes  193    146     55        16        44         249         869     7560       4120   800     1500  76       72      11.9
Albertson College                               Yes  587    479    158        38        62         678          41    13500       3335   500      675  67       73       9.4
Albertus Magnus College                         Yes  353    340    103        17        45         416         230    13290       5720   500     1500  90       93      11.5
Albion College                                  Yes 1899   1720    489        37        68        1594          32    13868       4826   450      850  89      100      13.7
Albright College                                Yes 1038    839    227        30        63         973         306    15595       4400   300      500  79       84      11.3
Alderson-Broaddus College                       Yes  582    498    172        21        44         799          78    10468       3380   660     1800  40       41      11.5
Alfred University                               Yes 1732   1425    472        37        75        1830         110    16548       5406   500      600  82       88      11.3
Allegheny College                               Yes 2652   1900    484        44        77        1707          44    17080       4440   400      600  73       91       9.9
Allentown Coll. of St. Francis de Sales         Yes 1179    780    290        38        64        1130         638     9690       4785   600     1000  60       84      13.3
Alma College                                    Yes 1267   1080    385        44        73        1306          28    12572       4552   400      400  79       87      15.3
Alverno College                                 Yes  494    313    157        23        46        1317        1235     8352       3640   650     2449  36       69      11.1
American International College                  Yes 1420   1093    220         9        22        1018         287     8700       4780   450     1400  78       84      14.7
Amherst College                                 Yes 4302    992    418        83        96        1593           5    19760       5300   660     1598  93       98       8.4
Anderson University                             Yes 1216    908    423        19        40        1819         281    10100       3520   550     1100  48       61      12.1
Andrews University                              Yes 1130    704    322        14        23        1586         326     9996       3090   900     1320  62       66      11.5
Antioch University                              Yes  713    661    252        25        44         712          23    15476       3336   400     1100  69       82      11.3
Aquinas College                                 Yes  619    516    219        20        51        1251         767    11208       4124   350     1615  55       65      12.7
Arkansas College (Lyon College)                 Yes  708    334    166        46        74         530         182     8644       3922   500      800  79       88      12.6
Assumption College                              Yes 2135   1700    491        23        59        1708         689    12000       5920   500      500  93       93      13.8
Augsburg College                                Yes  662    513    257        12        30        2074         726    11902       4372   540      950  65       65      12.8
Augustana College IL                            Yes 1879   1658    497        36        69        1950          38    13353       4173   540      821  78       83      12.7
Augustana College                               Yes  761    725    306        21        58        1337         300    10990       3244   600     1021  66       70      10.4
Austin College                                  Yes  948    798    295        42        74        1120          15    11280       4342   400     1150  81       95      13.0
Averett College                                 Yes  627    556    172        16        40         777         538     9925       4135   750     1350  59       67      22.4
Baker University                                Yes  602    483    206        21        47         958         466     8620       4100   400     2250  58       68      11.0
Baldwin-Wallace College                         Yes 1690   1366    662        30        61        2718        1460    10995       4410  1000     1000  68       74      17.6
Barat College                                   Yes  261    192    111        15        36         453         266     9690       4300   500      500  57       77       9.7
Bard College                                    Yes 1910    838    285        50        85        1004          15    19264       6206   750      750  98       98      10.4
Barnard College                                 Yes 2496   1402    531        53        95        2121          69    17926       8124   600      850  83       93      10.3
Barry University                                Yes  990    784    279        18        45        1811        3144    11290       5360   600     1800  76       78      12.6
Baylor University                               Yes 6075   5349   2367        34        66        9919         484     6450       3920   600     1346  71       76      18.5
Beaver College                                  Yes 1163    850    348        23        56         878         519    12850       5400   400      800  78       89      12.2
Bellarmine College                              Yes  807    707    308        39        63        1198         605     8840       2950   750     1290  74       82      13.1
Belmont Abbey College                           Yes  632    494    129        17        36         709         131     9000       4850   300     2480  78       85      13.2
Belmont University                              Yes 1220    974    481        28        67        1964         623     7800       3664   650      900  61       61      11.1
Beloit College                                  Yes 1320    923    284        26        54        1085          81    16304       3616   355      715  87       95      11.1
Benedictine College                             Yes  632    620    222        14        24         702         501     9550       3850   350      250  64       84      14.1
Bennington College                              Yes  519    327    114        25        53         457           2    21700       4100   600      500  35       59      10.1
Bentley College                                 Yes 3466   2330    640        20        60        3095        1533    13800       5510   630      850  87       87      17.5
Berry College                                   Yes 1858   1221    480        37        68        1620          49     8050       3940   350     2375  80       80      16.3
Bethany College                                 Yes  878    816    200        16        41         706          62     8740       3363   550     1700  62       68      11.6
Bethel College KS                               Yes  202    184    122        19        42         537         101     8540       3580   500     1400  61       80       8.8
Bethel College                                  Yes  502    384    104        11        28         347          74     6200       2900   600      800  63       63      11.7
Bethune Cookman College                         Yes 1646   1150    542        12        30        2128          82     5188       3396   650     2500  48       48      13.8
Birmingham-Southern College                     Yes  805    588    287        67        88        1376         207    11660       4325   400      900  74       79      14.0
Blackburn College                               Yes  500    336    156        25        55         421          27     6500       2700   500     1000  76       76      14.3
Bluefield College                               Yes  377    358    181        15        30         653         129     7150       4350   450     1500  61       67      17.8
Bluffton College                                Yes  692    514    209        20        50         760          81     9900       3990   400      900  76       71      13.3
                                            perc.alumni Expend Grad.Rate grad_rate_class
Abilene Christian University                     12   7041        60          Medium
Adelphi University                               16  10527        56          Medium
Adrian College                                   30   8735        54          Medium
Agnes Scott College                              37  19016        59          Medium
Alaska Pacific University                         2  10922        15            <NA>
Albertson College                                11   9727        55          Medium
Albertus Magnus College                          26   8861        63          Medium
Albion College                                   37  11487        73             Low
Albright College                                 23  11644        80             Low
Alderson-Broaddus College                        15   8991        52          Medium
Alfred University                                31  10932        73             Low
Allegheny College                                41  11711        76             Low
Allentown Coll. of St. Francis de Sales          21   7940        74             Low
Alma College                                     32   9305        68          Medium
Alverno College                                  26   8127        55          Medium
American International College                   19   7355        69          Medium
Amherst College                                  63  21424       100             Low
Anderson University                              14   7994        59          Medium
Andrews University                               18  10908        46          Medium
Antioch University                               35  42926        48          Medium
Aquinas College                                  25   6584        65          Medium
Arkansas College (Lyon College)                  24  14579        54          Medium
Assumption College                               30   7100        88             Low
Augsburg College                                 31   7836        58          Medium
Augustana College IL                             40   9220        71             Low
Augustana College                                30   6871        69          Medium
Austin College                                   33  11361        71             Low
Averett College                                  11   6523        48          Medium
Baker University                                 21   6136        65          Medium
Baldwin-Wallace College                          20   8086        85             Low
Barat College                                    35   9337        71             Low
Bard College                                     30  13894        79             Low
Barnard College                                  33  12580        91             Low
Barry University                                 11   9084        72             Low
Baylor University                                38   7503        72             Low
Beaver College                                   30   8954        73             Low
Bellarmine College                               31   6668        84             Low
Belmont Abbey College                            10   7550        52          Medium
Belmont University                               19   7614        49          Medium
Beloit College                                   26  12957        69          Medium
Benedictine College                              18   5922        58          Medium
Bennington College                               33  16364        55          Medium
Bentley College                                  20  10941        82             Low
Berry College                                    17  10511        63          Medium
Bethany College                                  29   7718        48          Medium
Bethel College KS                                32   8324        56          Medium
Bethel College                                   13   7623        35            High
Bethune Cookman College                           9   6817        58          Medium
Birmingham-Southern College                      34   8649        72             Low
Blackburn College                                53   8377        51          Medium
Bluefield College                                 3   6259        53          Medium
```

[[2]]

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal
Angelo State University	No	3540	2001	1016	24	54	4190	1512	5130	3592	500	2000	60	62
Appalachian State University	No	7313	4664	1910	20	63	9940	1035	6806	2540	96	2000	83	96
Arizona State University Main campus	No	12809	10308	3761	24	49	22593	7585	7434	4850	700	2100	88	93
Arkansas Tech University	No	1734	1729	951	12	52	3602	939	3460	2650	450	1000	57	60
Auburn University-Main Campus	No	7548	6791	3070	25	57	16262	1716	6300	3933	600	1908	85	91
Bemidji State University	No	1208	877	546	12	36	3796	824	4425	2700	660	1800	57	62
Bloomsburg Univ. of Pennsylvania	No	6773	3028	1025	15	55	5847	946	7844	2948	500	1680	66	68
Bowling Green State University	No	9251	7333	3076	14	45	13699	1213	7452	3352	600	1700	81	89
California Polytechnic-San Luis	No	7811	3817	1650	47	73	12911	1404	7380	4877	612	2091	72	81
California State University at Fresno	No	4540	3294	1483	5	60	13494	1254	7706	4368	600	1926	90	90
Castleton State College	No	1257	940	363	9	22	1547	294	7656	4690	400	700	89	91
Central Connecticut State University	No	4158	2532	902	6	24	6394	3881	5962	4444	500	985	69	73
Central Missouri State University	No	4681	4101	1436	10	35	8094	1596	4620	3288	300	2250	69	80
Central Washington University	No	2785	2011	1007	8	65	6507	898	7242	3603	654	1416	67	89
Christopher Newport University	No	883	766	428	3	37	2910	1749	7860	4750	525	1889	80	82
Clemson University	No	8065	5257	2301	37	65	11755	770	8116	3610	800	1618	82	88
Clinch Valley Coll. of the Univ. of Virginia	No	689	561	250	15	30	1125	422	7168	3689	600	1900	67	67
College of Charleston	No	4772	3140	1265	22	55	6851	1200	6120	3460	666	2316	73	78
College of William and Mary	No	7117	3106	1217	68	88	5186	134	11720	4298	600	800	89	92
Colorado State University	No	9478	6312	2194	29	65	15646	1829	8412	4180	470	1800	87	89
Delta State University	No	967	945	459	15	48	2806	538	4528	1880	500	1200	49	63
Dickinson State University	No	434	412	319	10	30	1376	237	4486	2146	600	2000	50	64
East Carolina University	No	9274	6362	2435	14	44	13171	1687	7248	3240	500	1700	74	78
East Tennessee State University	No	3330	2730	1303	15	36	6706	2640	5800	3000	600	2200	73	75
Eastern Connecticut State University	No	2172	1493	564	14	50	2766	1531	5962	4316	650	500	71	76
Eastern Illinois University	No	5597	4253	1565	12	38	9161	845	5710	3066	120	1730	62	71
Emporia State University	No	1256	1256	853	43	79	3957	588	5401	3144	450	1888	72	75
Evergreen State College	No	1801	1101	438	14	50	3065	363	6297	4600	600	1323	75	78
Fayetteville State University	No	1455	1064	452	1	16	2632	617	6806	2550	350	766	75	75
Florida International University	No	3306	2079	1071	42	89	10208	9310	6597	2494	800	3028	81	96
Florida State University	No	11651	8683	3023	50	90	18906	3242	6680	4060	600	1020	80	89
Fort Lewis College	No	3440	2823	1123	16	35	3793	486	6198	3320	500	2500	89	97
Francis Marion University	No	1801	1655	819	13	38	3224	436	5840	3138	400	2430	76	76
George Mason University	No	5653	4326	1727	17	29	9528	3822	10800	4840	580	1050	93	96
Georgia Institute of Technology	No	7837	4527	2276	89	99	8528	654	6489	4438	795	1164	92	92
Georgia State University	No	3793	2341	1238	9	24	7732	9054	6744	2655	720	3450	87	89
Grand Valley State University	No	5165	3887	1561	20	60	8234	2619	6108	3800	500	1000	64	66
Illinois State University	No	8681	6695	2408	10	35	15701	1823	7799	3403	537	2605	77	84
Indiana State University	No	5659	4761	3147	10	31	8596	1949	6892	3706	600	2500	72	76
Indiana University at Bloomington	No	16587	13243	5873	25	72	24763	2717	9766	3990	600	2000	77	88
Iowa State University	No	8427	7424	3441	26	59	18676	1715	7550	3224	640	2055	81	88
James Madison University	No	11223	5285	2082	32	72	9652	742	7994	4544	500	732	77	81
Jersey City State College	No	2957	1423	691	10	30	3817	1394	3946	4800	400	1500	63	67
Johnson State College	No	833	669	279	3	13	1224	345	7656	4690	500	624	80	91
Kansas State University	No	5880	4075	2833	25	55	14914	2246	6995	3120	600	2000	76	86
Keene State College	No	3121	2446	822	5	19	3480	776	7870	4157	500	1150	73	73
Lamar University	No	2336	1725	1043	10	27	5438	4058	4752	3040	508	1463	48	82
Lander University	No	1166	1009	510	9	33	2074	341	4938	2987	528	1702	67	77
Lincoln University	No	1660	1091	326	15	41	1196	33	4818	3400	350	1400	71	72
Lock Haven University of Pennsylvania	No	3570	2215	651	17	41	3390	325	7352	3620	225	500	47	55
Longwood College	No	2747	1870	724	12	47	2874	118	7920	3962	550	2200	74	80

	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate	Room.Board	Books	Personal	PhD	Terminal
Lock Haven University of Pennsylvania	No	3570	2215	651	17	41	3390	325	7352	3620	225	500	47	55
Longwood College	No	2747	1870	724	12	47	2874	118	7920	3962	550	2200	74	80
Louisiana State University at Baton Rouge	No	5996	4993	3079	29	57	16269	3757	5925	2980	600	2242	69	77

	S.F.Ratio	perc.alumni	Expend	Grad.Rate	grad_rate_class
Angelo State University	23.1	5	4010	34	High
Appalachian State University	18.3	14	5854	70	Medium
Arizona State University Main campus	18.9	5	4602	48	Medium
Arkansas Tech University	19.6	5	4739	48	Medium
Auburn University-Main Campus	16.7	18	6642	69	Medium
Bemidji State University	19.6	16	3752	46	Medium
Bloomsburg Univ. of Pennsylvania	18.0	19	7041	75	Low
Bowling Green State University	21.1	14	6918	67	Medium
California Polytechnic-San Luis	19.8	13	8453	59	Medium
California State University at Fresno	21.2	8	7268	61	Medium
Castleton State College	14.7	8	6318	79	Low
Central Connecticut State University	16.7	4	4900	49	Medium
Central Missouri State University	19.7	4	5501	50	Medium
Central Washington University	18.1	0	6413	51	Medium
Christopher Newport University	21.2	16	4639	48	Medium
Clemson University	18.0	17	7597	73	Low
Clinch Valley Coll. of the Univ. of Virginia	18.1	9	4417	46	Medium
College of Charleston	17.2	18	4776	51	Medium
College of William and Mary	12.1	31	9534	93	Low
Colorado State University	19.2	10	7850	59	Medium
Delta State University	17.1	16	5113	58	Medium
Dickinson State University	16.5	28	4525	46	Medium
East Carolina University	13.2	18	9002	58	Medium
East Tennessee State University	14.0	9	9825	42	Medium
Eastern Connecticut State University	16.9	14	5719	50	Medium
Eastern Illinois University	16.2	5	5682	76	Low
Emporia State University	19.3	4	5527	50	Medium
Evergreen State College	18.1	14	8355	68	Medium
Fayetteville State University	15.1	10	6972	24	High
Florida International University	13.9	20	6722	66	Medium
Florida State University	23.1	15	7250	58	Medium
Fort Lewis College	19.1	6	4362	46	Medium
Francis Marion University	19.1	8	5039	43	Medium
George Mason University	19.3	7	6751	46	Medium
Georgia Institute of Technology	19.3	33	11271	70	Medium
Georgia State University	19.0	10	7762	34	High
Grand Valley State University	20.6	9	5063	57	Medium
Illinois State University	21.0	16	5569	54	Medium

Q2)

i)We use the load() function to load the marketing.RData dataset, and use the dim() function to view the shape of the dataset. The summary function below gives us the statistical overview of the dataset.

Next we shall use the na.omit() function that removes all incomplete cases of the marketing data frame. Next when we use the dim() we see that the shape is now changed with 6876 records and 14 variables.

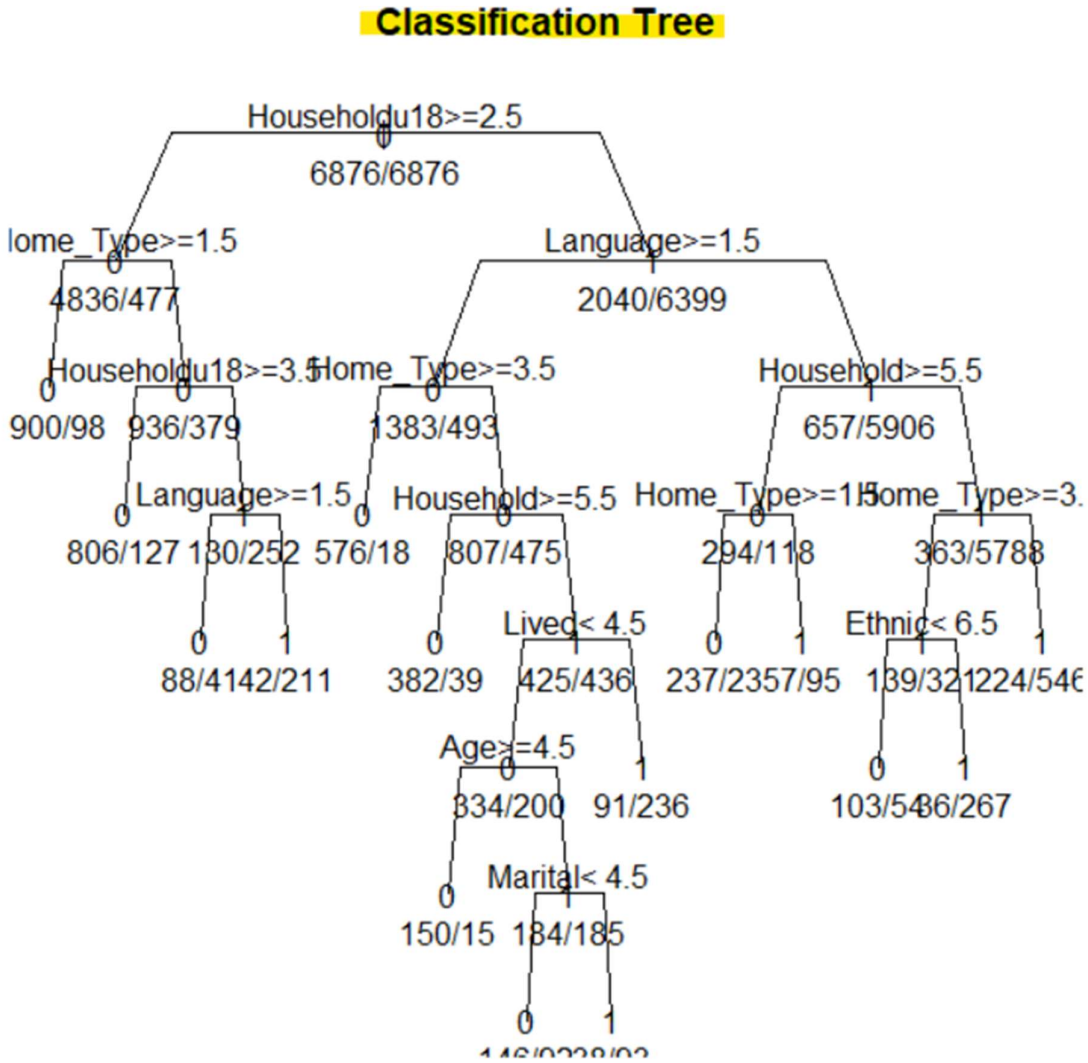
```
> library(randomForest)
> library("rpart") #install.packages("rpart")
> load("marketing.RData")
>
> dim(marketing) # 8993 x 14
[1] 8993   14
> summary(marketing)
     Income           Sex           Marital          Age             Edu          Occupation       Lived        Dual_Income     Household      Householdu18
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :0.0000
 1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:1.000   1st Qu.:4.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:0.0000
 Median :5.000   Median :2.000   Median :3.000   Median :3.000   Median :4.000   Median :4.000   Median :5.000   Median :1.000   Median :3.000   Median :0.0000
 Mean   :4.895   Mean   :1.547   Mean   :3.031   Mean   :3.415   Mean   :3.835   Mean   :3.788   Mean   :4.198   Mean   :1.545   Mean   :2.852   Mean   :0.6669
 3rd Qu.:7.000   3rd Qu.:2.000   3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:1.0000
 Max.   :9.000   Max.   :2.000   Max.   :5.000   Max.   :7.000   Max.   :6.000   Max.   :9.000   Max.   :5.000   Max.   :3.000   Max.   :9.000   Max.   :9.0000
                                 NA's   :160                     NA's   :86      NA's   :136     NA's   :913                     NA's   :375
     Status        Home_Type        Ethnic         Language
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:1.000   1st Qu.:1.000   1st Qu.:5.000   1st Qu.:1.000
 Median :2.000   Median :1.000   Median :7.000   Median :1.000
 Mean   :1.837   Mean   :1.856   Mean   :5.956   Mean   :1.127
 3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:7.000   3rd Qu.:1.000
 Max.   :3.000   Max.   :5.000   Max.   :8.000   Max.   :3.000
 NA's   :240     NA's   :357     NA's   :68      NA's   :359
>
> marketing <- na.omit(marketing)
> dim(marketing) # 6876 x 14
[1] 6876   14
>
> randomized_data <- marketing # for question i)
> permuted_data <- marketing # for question ii)
```

Here we use the sample() function to generate randomised values.

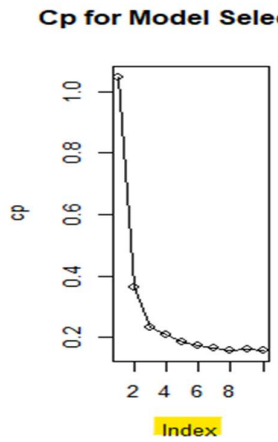
```
for (i in 1:14) {
  min_val = min(marketing[,i])
  max_val = max(marketing[,i])
  randomized_data[,i] = floor(runif(nrow(marketing), min=min_val, max=max_val + 1))
  permuted_data[,i] = sample(min_val: max_val, nrow(marketing), replace = TRUE)
}

marketing$result <- 1
randomized_data$result <- 0
permuted_data$result <- 0

new_data_1 <- rbind(marketing, randomized_data)
new_data_2 <- rbind(marketing, permuted_data)
```

Here we keep the minsplit as 350 and we get the following classification tree when plotted using plot() function.

```
model <- rpart.control(minsplit = 350, xval = 40, cp = 0)
classif <- rpart(result~., data = new_data_1, method = "class", control = model)

x11()
plot(classif, branch = .8, uniform = T, compress = T, main="Classification Tree")

text(classif, use.n = T, all = T, cex = 1)

summary(classif )
```

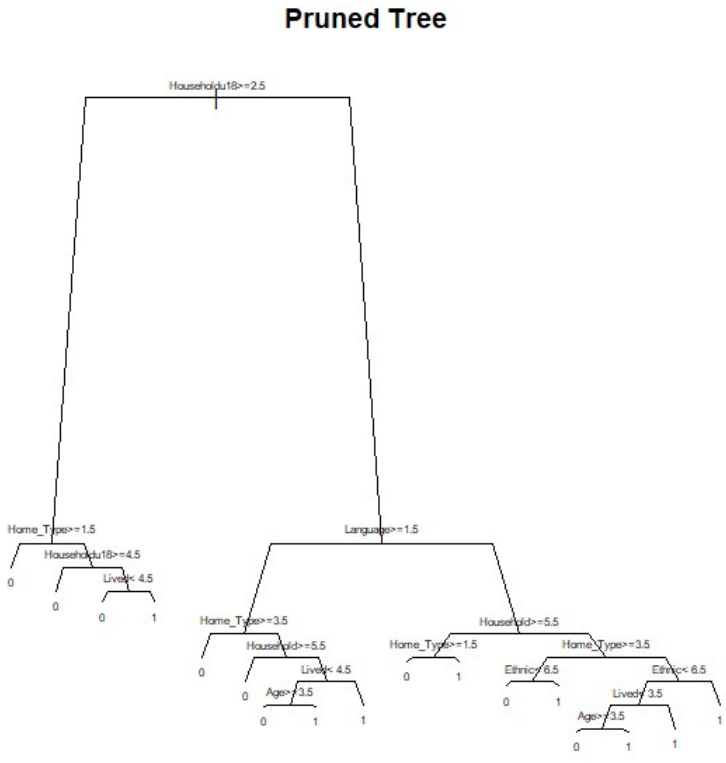



```
plot(classif$cptable[,4], type = "o", main = "Cp for Model Selection", ylab="cp")
```



Next we generate a pruned tree

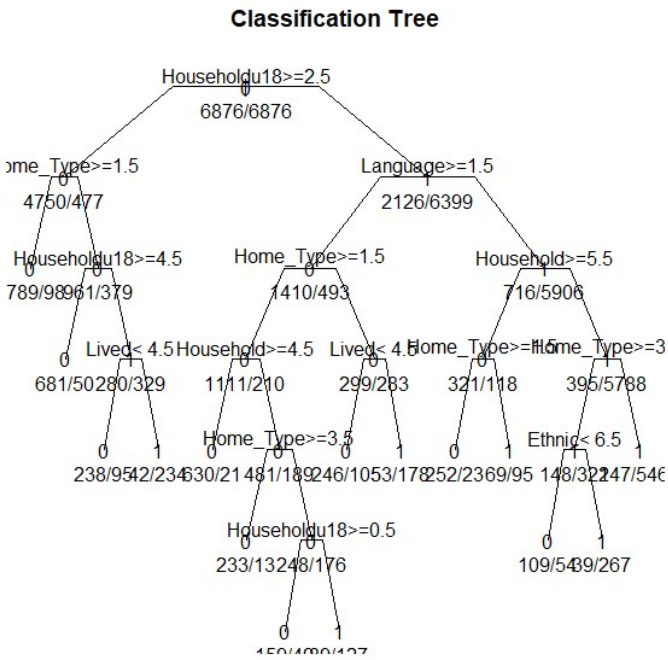
```
# Pruning
min_cp = which.min(classif$cptable[,5])
pruned_fit <- prune(classif, cp = classif$cptable[min_cp,1])

x11()
plot(pruned_fit, branch = 0.8, compress = T, main = "Pruned Tree")
text(pruned_fit, cex = 0.5)
```



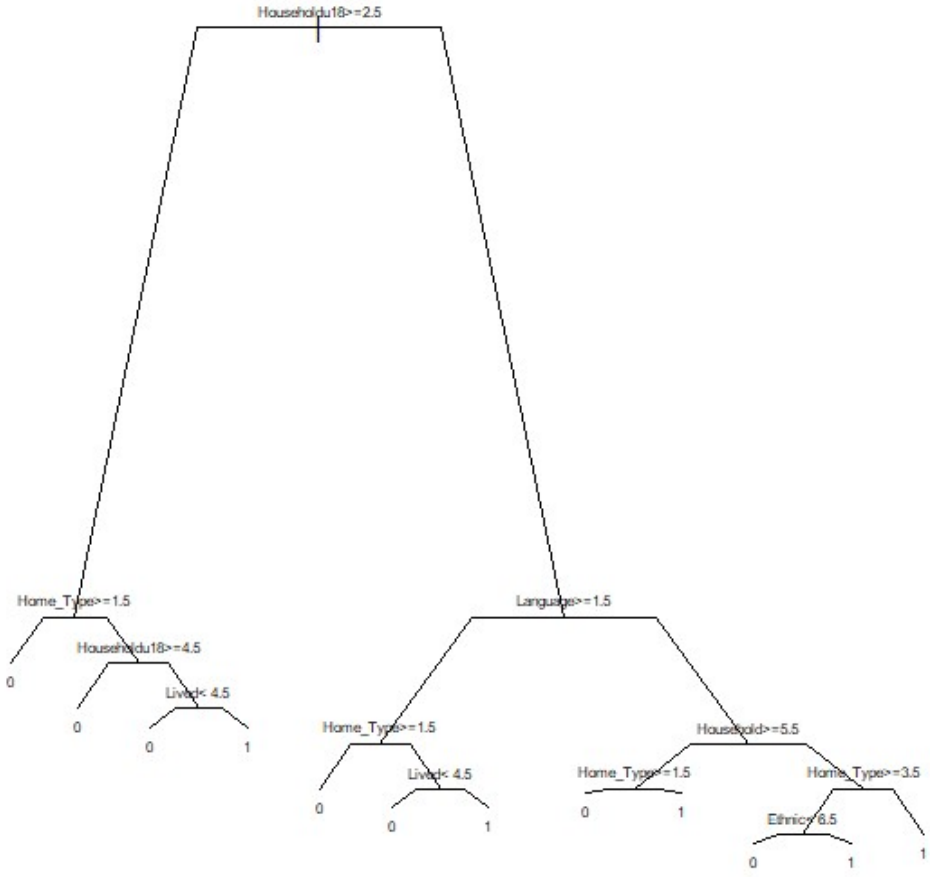
ii)

```
model.control <- rpart.control(minsplit = 5, xval = 10, cp = 0)
classif <- rpart(result~., data = new_data_2, method = "class", control = model.control)

x11()
plot(classif, branch = .4, uniform = T, compress = T, main="Classification Tree")
text(classif, use.n = T, all = T, cex = 1)

# Pruning
min_cp = which.min(classif$cptable[,4])
pruned_fit <- prune(classif, cp = classif$cptable[min_cp,1])

x11()
plot(pruned_fit, branch = 0.3, compress = T, main = "Pruned Tree")
text(pruned_fit, cex = 0.5)
```

```
model.control <- rpart.control(minsplit = 350, xval = 10, cp = 0)
classif <- rpart(result~., data = new_data_2, method = "class", control = model.control)

x11()
plot(classif, branch = .4, uniform = T, compress = T, main="Classification Tree")
text(classif, use.n = T, all = T, cex = 1)
```



```
# Pruning
min_cp = which.min(classif$cptable[,4])
pruned_fit <- prune(classif, cp = classif$cptable[min_cp,1])

x11()
plot(pruned_fit, branch = 0.5, compress = T, main = "Pruned Tree")
text(pruned_fit, cex = 0.5)
```



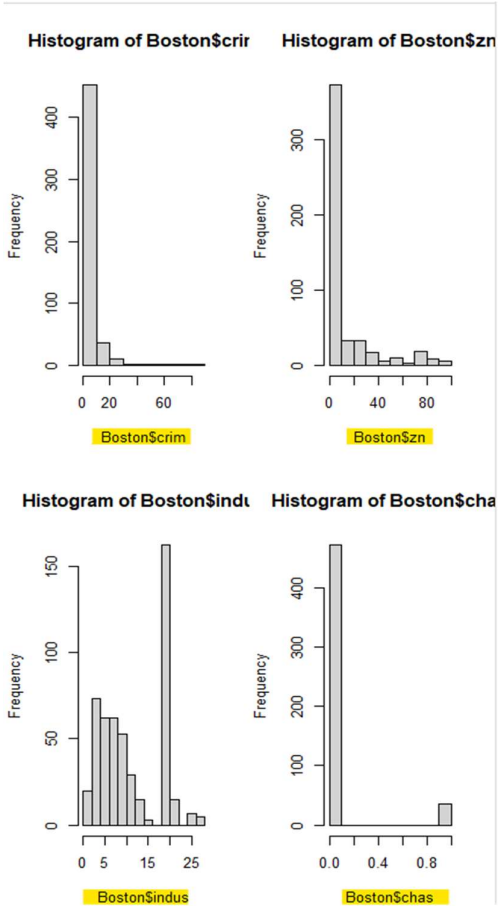
Q3

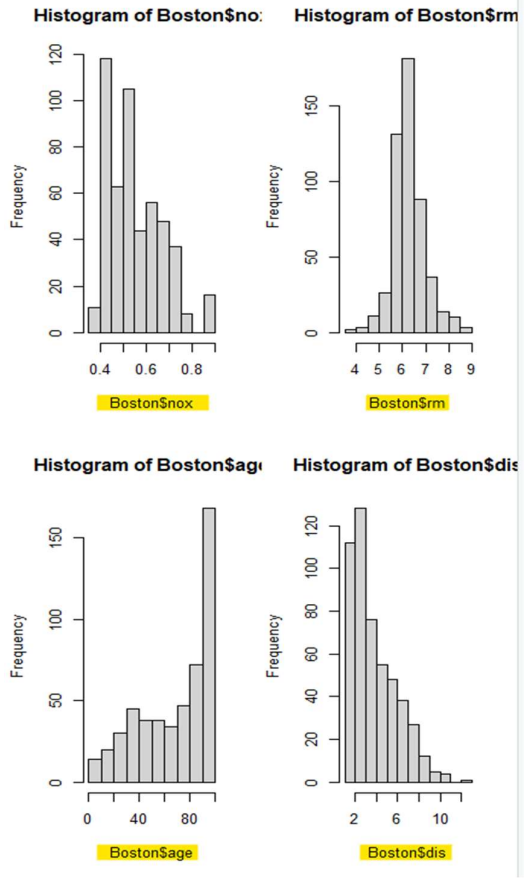
) a) Here we use setwd() to set the working directory and use the head function to view till tuple number 5, i.e, 6 rows in the dataset. We check the shape of the dataset using the dim()function and we see that the data set is comprised of 506 tuples and 13 variables. We further move forward to store the data set into a dataframe variable.

```
> rm(list=ls())
> library(arules)
> setwd("/Users/Sriram/Desktop/R projects/Sem 2/Hw1/")
> library(ISLR2)
> head(Boston)
     crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
> dim(Boston)
[1] 506  13
> ?Boston
> Boston_data<-Boston
```

Next we shall view all the histograms of the variables in the dataframe, using the hist() function.

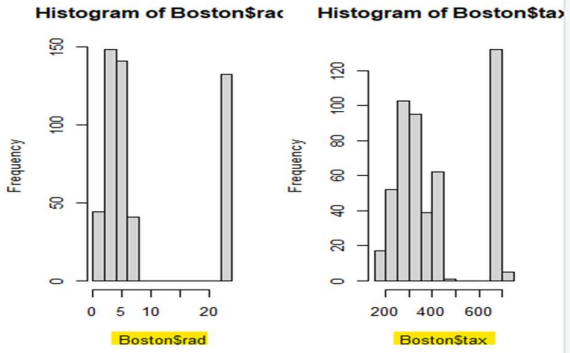
```
par(mfrow = c(2,2))
hist(Boston$crim)
hist(Boston$zn)
hist(Boston$indus)
hist(Boston$chas)
hist(Boston$nox)
hist(Boston$rm)
hist(Boston$age)
hist(Boston$dis)
hist(Boston$rad)
hist(Boston$tax)
hist(Boston$ptratio)
hist(Boston$lstat)
hist(Boston$medv)
```

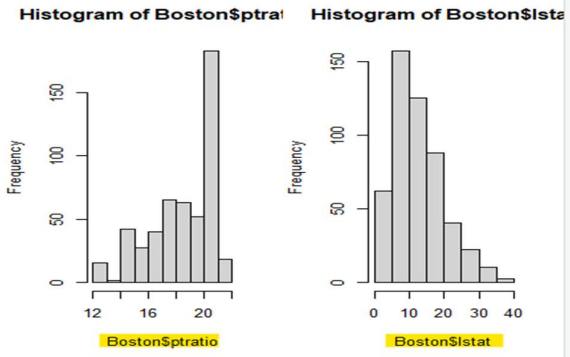

Histogram of Boston$crir
Histogram of Boston$zn
Frequency
Boston$crim
Boston$zn

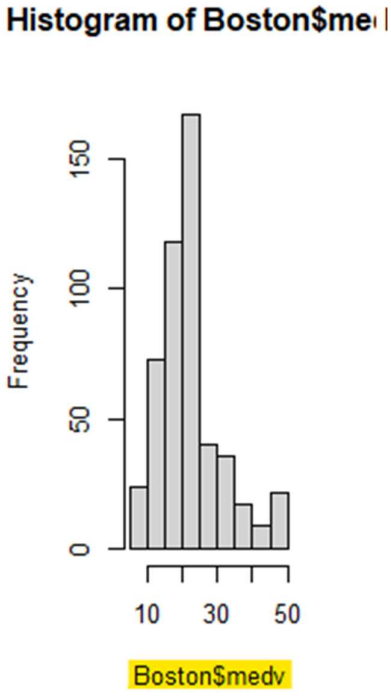

Histogram of Boston$no:
Histogram of Boston$rm
Frequency
Boston$nox
Boston$rm

Histogram of Boston$indu
Histogram of Boston$cha
Frequency
Boston$indus
Boston$chas

Histogram of Boston$ag
Histogram of Boston$dis
Frequency
Boston$age
Boston$dis

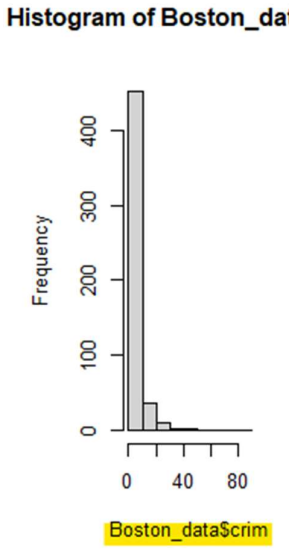

Histogram of Boston$rac
Histogram of Boston$tax
Frequency
Boston$rad
Boston$tax

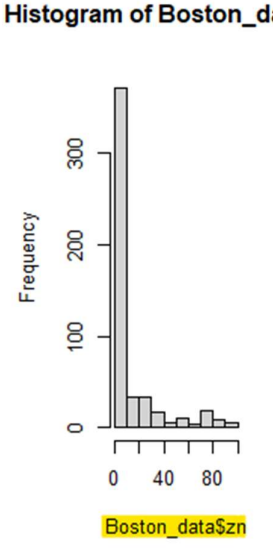

Histogram of Boston$ptrat
Histogram of Boston$lsta
Frequency
Boston$ptratio
Boston$lstat

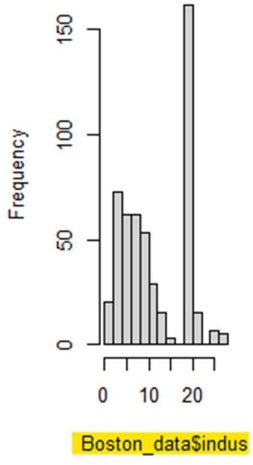

Histogram of Boston$me
Frequency
Boston$medv

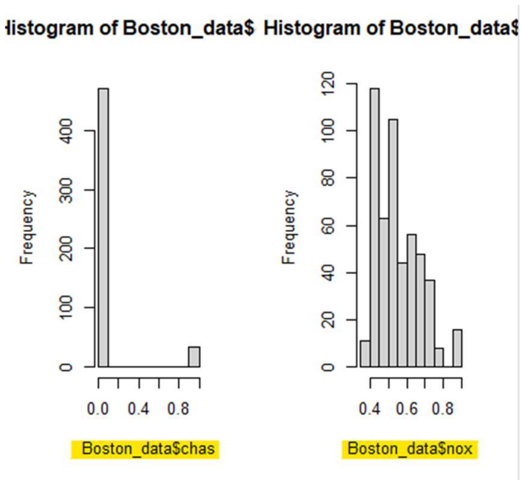
Next we plot box plots of each variables and add labels to it. Below you can see the boxplot's and hist() of the following variable.

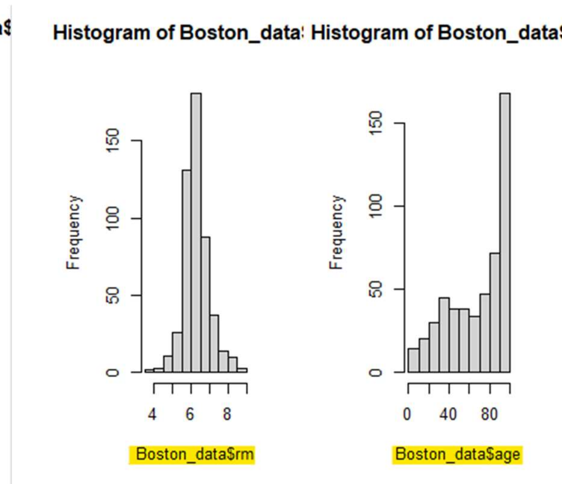
```
> hist(Boston_data$crim)
>
> boxplot.stats(Boston_data$crim)$stats
[1] 0.00632 0.08199 0.25651 3.67822 8.98296
> Boston_data$crim = ordered(cut(Boston_data[["crim"]],c(10,20,40,60,100)), labels = c('low-Threat','Medium-Threat','High-alert','CodeRed'))
>
>
> hist(Boston_data$zn)
> Boston_data[["zn"]]<- ordered (cut(Boston_data[["zn"]], c (0,15,50,100)), labels= c("no-zone", "less-Zone", "High-Zone"))
>
> hist(Boston_data$indus)
> boxplot.stats(Boston_data$indus)$stats
[1]  0.46  5.19  9.69 18.10 27.74
> Boston_data$indus = ordered(cut(Boston_data[["indus"]],c(0,10,20,40)), labels = c('Low','Moderate','High'))
>
> hist(Boston_data$chas)
> Boston_data[["chas"]]<-ordered(Boston_data$chas,labels=c(" Not bound to river","bound to river"))
>
>
> hist(Boston_data$nox)
> boxplot.stats(Boston_data$nox)$stats
[1] 0.385 0.449 0.538 0.624 0.871
> Boston_data$nox = ordered(cut(Boston_data[["nox"]],c(0.1,0.3,0.4,0.8)), labels = c('Low-Concentration','Moderate-Concentration','High-Concentration'))
Error in factor(x, ..., ordered = TRUE) :
  invalid 'labels'; length 3 should be 1 or 2
>
>
> hist(Boston_data$rm)
> boxplot.stats(Boston_data$rm)$stats
[1] 4.8800 5.8850 6.2085 6.6250 7.6910
> Boston_data$rm = ordered(cut(Boston_data[["rm"]],c(0,4,7,9)), labels = c('Small houses','Medium Houses','Big Houses'))
>
>
> hist(Boston_data$age)
> boxplot.stats(Boston_data$age)$stats
[1]   2.9  45.0  77.5  94.1 100.0
> Boston_data$age = ordered(cut(Boston_data[["age"]],c(0,25,50,70,100)), labels = c('Silver','Gold','Diamond','Too old'))
```





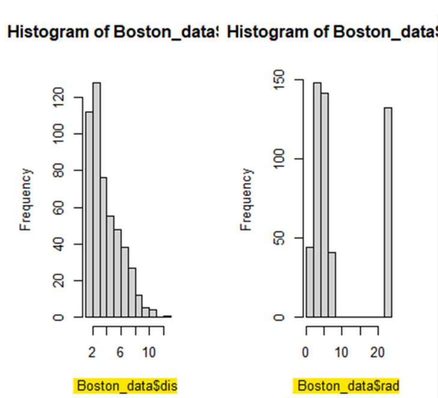


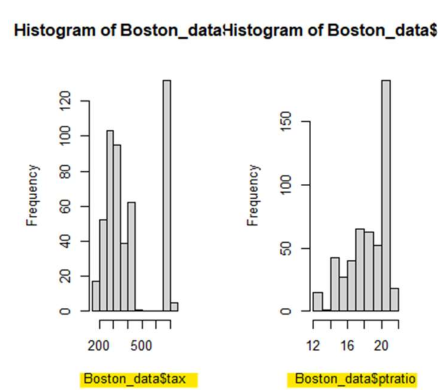




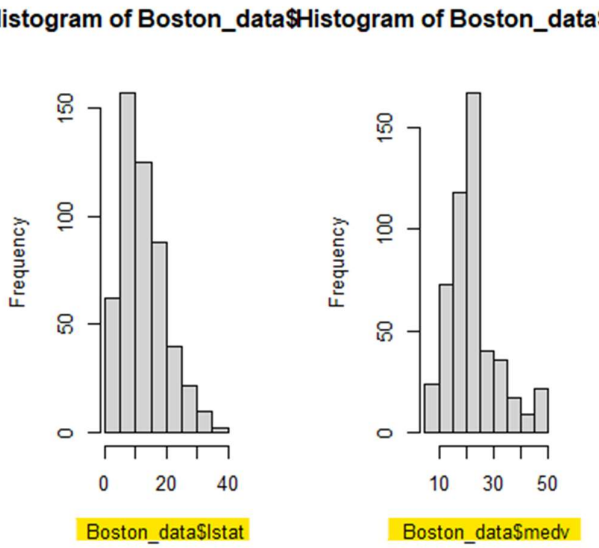

```
Boston_data$age = ordered(cut(Boston_data[["age"]],c(0,25,50,70,100)), labels = c('Silver','Gold','Diamond','Too old'))
Boston_data$age
  [1] Diamond Too old Diamond Gold    Diamond Diamond Diamond Too old Too old Too old Too old Too old Gold    Diamond Too old Diamond Gold    Too old Gold    Diamond Too old
 [22] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Diamond Diamond Gold    Gold    Silver  Silver  Silver
 [43] Silver  Silver  Gold    Gold    Gold    Too old Too old Diamond Gold    Diamond Silver  Silver  Gold    Silver  Gold    Gold    Gold    Gold    Diamond Too old Diamond
 [64] Gold    Diamond Silver  Gold    Silver  Gold    Gold    Silver  Silver  Silver  Silver  Silver  Gold    Too old Gold    Diamond Gold    Gold    Too old Gold    Gold
 [85] Gold    Diamond Gold    Diamond Too old Diamond Diamond Too old Diamond Gold    Too old Diamond Diamond Too old Gold    Diamond Too old Too old Too old Too old Too old
[106] Too old Too old Too old Too old Too old Diamond Too old Too old Too old Too old Too old Too old Too old Too old Diamond Diamond Too old Too old Too old Too old Too old
[127] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[148] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[169] Too old Too old Too old Too old Too old Too old Diamond Gold    Gold    Too old Too old Diamond Too old Diamond Too old Too old Too old Diamond Diamond Gold    Gold
[190] Gold    Silver  Gold    Gold    Silver  Silver  Gold    Gold    Gold    Gold    Silver  Silver  Gold    Silver  Gold    Gold    Silver  Diamond Too old Diamond Too old
[211] Too old Too old Diamond Gold    Silver  Gold    Diamond Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Silver  Silver  Diamond
[232] Too old Too old Too old Diamond Diamond Too old Too old Silver  Gold    Diamond Diamond Diamond Silver  Too old Too old Gold    Too old Gold    Silver  Silver  Silver
[253] Silver  Silver  Gold    Silver  Gold    Too old Too old Too old Too old Too old Too old Too old Too old Diamond Too old Diamond Diamond Diamond Gold    Silver  Diamond
[274] Diamond Gold    Gold    Gold    Gold    Gold    Gold    Diamond Gold    Gold    Silver  Silver  Gold    Gold    Gold    Gold    Silver  Gold    Gold    Silver  Silver
[295] Gold    Gold    Diamond Diamond Silver  Silver  Gold    Gold    Silver  Silver  Gold    Diamond Too old Too old Too old Too old Gold    Diamond Too old Too old Too old
[316] Too old Too old Too old Diamond Diamond Diamond Diamond Gold    Too old Gold    Silver  Gold    Gold    Gold    Silver  Gold    Gold    Silver  Gold    Gold    Gold
[337] Gold    Diamond Gold    Gold    Diamond Gold    Diamond Diamond Gold    Gold    Diamond Gold    Gold    Gold    Gold    Gold    Silver  Gold    Silver  Silver  Too old
[358] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[379] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[400] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[421] Too old Too old Too old Too old Too old Too old Diamond Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[442] Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old Too old
[463] Too old Too old Diamond Gold    Too old Too old Too old Diamond Too old Too old Too old Diamond Too old Too old Too old Too old Too old Too old Diamond Too old Too old
[484] Gold    Gold    Diamond Too old Diamond Too old Too old Too old Too old Too old Diamond Gold    Gold    Too old Too old Diamond Too old Too old Diamond Too old Too old
[505] Too old Too old
Levels: Silver < Gold < Diamond < Too old
```

```
hist(Boston_data$dis)
boxplot.stats(Boston_data$dis)$stats
[1] 1.12960 2.10000 3.20745 5.21190 9.22290
Boston_data$dis = ordered(cut(Boston_data[["dis"]],c(0,4,7,14)), labels = c('Near','Far','Very Far'))
hist(Boston_data$rad)
boxplot.stats(Boston_data$rad)$stats
[1]  1  4  5 24 24
Boston_data$rad = ordered(cut(Boston_data[["rad"]],c(0,2,10,26)), labels = c('Less Accesible','Moderately Accesible','Highly Accesible'))
hist(Boston_data$tax)
boxplot.stats(Boston_data$tax)$stats
[1] 187 279 330 666 711
Boston_data$tax = ordered(cut(Boston_data[["tax"]],c(187,279,660,711)), labels = c('Low','Moderate','High'))
hist(Boston_data$ptratio)
boxplot.stats(Boston_data$ptratio)$stats
[1] 13.60 17.40 19.05 20.20 22.00
Boston_data$ptratio = ordered(cut(Boston_data[["ptratio"]],c(10,19,26)), labels = c('Low','High'))
```





```
hist(Boston_data$lstat)
boxplot.stats(Boston_data$lstat)$stats
1]  1.73  6.93 11.36 16.96 31.99
Boston_data$lstat = ordered(cut(Boston_data[["lstat"]],c(0,12,40)), labels = c('Low','High'))
hist(Boston_data$medv)
boxplot.stats(Boston_data$medv)$stats
1]  5.0 17.0 21.2 25.0 37.0
Boston_data$medv = ordered(cut(Boston_data[["medv"]],c(0,22,51)), labels = c('Low','High'))
```



b) Next we use the apriori() in order to calculate the apriori rules. We see that here we obtain 50 support counts.

```
> incedence_matrix = as(Boston_data,"transactions")
Warning message:
Column(s) 5 not logical or factor. Applying default discretization (see '? discretizeDF').
> incedence_matrix
transactions in sparse format with
 506 transactions (rows) and
 37 items (columns)
> itemFrequencyPlot(incedence_matrix,support=0.001,cex.name= 0.8)
> rules<-apriori(incedence_matrix,parameter= list(support = 0.1,confidence = 0.3))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.3    0.1    1 none FALSE            TRUE       5     0.1      1     10  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 50

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[37 item(s), 506 transaction(s)] done [0.00s].
sorting and recoding items ... [25 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.00s].
writing ... [17755 rule(s)] done [0.01s].
creating S4 object  ... done [0.01s].
```

The summary() function is used to view the statistical description of the apriori rules.

```
> summary(rules)
set of 17755 rules

rule length distribution (lhs + rhs):sizes
   1    2    3    4    5    6    7    8    9   10
  17  311 1616 3564 4283 3588 2443 1328  495  110

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   4.000   5.000   5.405   6.000  10.000

summary of quality measures:
    support         confidence        coverage           lift            count
 Min.   :0.1008   Min.   :0.3000   Min.   :0.1008   Min.   :0.4699   Min.   : 51.00
 1st Qu.:0.1522   1st Qu.:0.8104   1st Qu.:0.1838   1st Qu.:1.1500   1st Qu.: 77.00
 Median :0.1877   Median :0.9277   Median :0.2095   Median :1.6382   Median : 95.00
 Mean   :0.1871   Mean   :0.8597   Mean   :0.2266   Mean   :1.7839   Mean   : 94.69
 3rd Qu.:0.2115   3rd Qu.:0.9778   3rd Qu.:0.2431   3rd Qu.:2.0220   3rd Qu.:107.00
 Max.   :0.9308   Max.   :1.0000   Max.   :1.0000   Max.   :3.8333   Max.   :471.00

mining info:
            data ntransactions support confidence                                                                                  call
 incedence_matrix           506     0.1        0.3 apriori(data = incedence_matrix, parameter = list(support = 0.1, confidence = 0.3))
```

c) Next we see the rules of medv being low with respect to dis being near, where lift is greater than 1.2 . We also use the inspect() function to view the support confidence and lift values, and head() function with n =3, i.e., to view the first 3 records .

```
> #3c
> rules_medv = subset(rules, subset = lhs %in% "dis=Near" & rhs %in% "medv=Low" & lift>1.2)
> inspect(head(sort(rules_medv, by = "confidence"), n = 3))
    lhs
[1] {rm=Medium Houses, age=Too old, dis=Near, ptratio=High, lstat=High}
[2] {chas= Not bound to river, rm=Medium Houses, age=Too old, dis=Near, ptratio=High, lstat=High}
[3] {nox=[0.597,0.871], rm=Medium Houses, dis=Near, ptratio=High, lstat=High}
>
```

```
    rhs          support   confidence coverage  lift     count
=> {medv=Low} 0.2786561 0.9657534  0.2885375 1.720673 141
=> {medv=Low} 0.2727273 0.9650350  0.2826087 1.719393 138
=> {medv=Low} 0.2173913 0.9649123  0.2252964 1.719175 110
```

d)

```
> #3d
> rules_ptratio_low<-subset(rules,subset= rhs %in% "ptratio=Low" & lift>1.2)
> inspect(head(sort(rules_ptratio_low, by="confidence"),n=3))
    lhs                                                  rhs          support   confidence coverage  lift     count
[1] {indus=Moderate, dis=Near, rad=Moderately Accesible} => {ptratio=Low} 0.1067194 1.0000000  0.1067194 2.000000 54
[2] {indus=Low, rm=Big Houses}                           => {ptratio=Low} 0.1047431 0.9636364  0.1086957 1.927273 53
[3] {indus=Low, rm=Big Houses, medv=High}                => {ptratio=Low} 0.1047431 0.9636364  0.1086957 1.927273 53
>
```