***Problem statement :***

***Implement k-means in R, finding the clusters***

Step 1 : read file

> bank<- read.csv("C:/Users/Wanwan Zhang/Desktop/2016FALL/ADS/2/bank-data.csv", stringsAsFactors=FALSE)

Step 2 :　checking structure of the fetched data

> str(bank)

'data.frame': 600 obs. of 12 variables:

$ id : Factor w/ 600 levels "ID12101","ID12102",..: 1 2 3 4 5 6 7 8 9 10 ...

$ age : int 48 40 51 23 57 57 22 58 37 54 ...

$ sex : Factor w/ 2 levels "FEMALE","MALE": 1 2 1 1 1 1 2 2 1 2 ...

$ region : Factor w/ 4 levels "INNER\_CITY","RURAL",..: 1 4 1 4 2 4 2 4 3 4 ...

$ income : num 17546 30085 16575 20375 50576 ...

$ married : Factor w/ 2 levels "NO","YES": 1 2 2 2 2 2 1 2 2 2 ...

$ children : int 1 3 0 3 0 2 0 0 2 2 ...

$ car : Factor w/ 2 levels "NO","YES": 1 2 2 1 1 1 1 2 2 2 ...

$ save\_act : Factor w/ 2 levels "NO","YES": 1 1 2 1 2 2 1 2 1 2 ...

$ current\_act: Factor w/ 2 levels "NO","YES": 1 2 2 2 1 2 2 2 1 2 ...

$ mortgage : Factor w/ 2 levels "NO","YES": 1 2 1 1 1 1 1 1 1 1 ...

$ pep : Factor w/ 2 levels "NO","YES": 2 1 1 1 1 2 2 1 1 1 ...

> summary(bank)

id age sex region income married children

ID12101: 1 Min. :18.00 FEMALE:300 INNER\_CITY:269 Min. : 5014 NO :204 Min. :0.000

ID12102: 1 1st Qu.:30.00 MALE :300 RURAL : 96 1st Qu.:17265 YES:396 1st Qu.:0.000

ID12103: 1 Median :42.00 SUBURBAN : 62 Median :24925 Median :1.000

ID12104: 1 Mean :42.40 TOWN :173 Mean :27524 Mean :1.012

ID12105: 1 3rd Qu.:55.25 3rd Qu.:36173 3rd Qu.:2.000

ID12106: 1 Max. :67.00 Max. :63130 Max. :3.000

(Other):594

car save\_act current\_act mortgage pep

NO :304 NO :186 NO :145 NO :391 NO :326

YES:296 YES:414 YES:455 YES:209 YES:274

##list of rows with missing values

> bank[!complete.cases(bank),]

[1] id age sex region income married

[7] children car save\_act current\_act mortgage pep

<0 rows> (or 0-length row.names)

> bank[,!complete.cases(bank)]

data frame with 0 columns and 600 rows

> bank<-na.omit(bank,na.action = TRUE)

Step 3 : convert the factor value to numeric

> ##select only numeric data and removing id column

bank[,3][bank[,3] == "MALE"] = 1

bank[,3][bank[,3] == "FEMALE"] = 2

as.numeric(bank[,3])

bank[,4][bank[,4] == "INNER\_CITY"] = 1

bank[,4][bank[,4] == "TOWN"] = 2

bank[,4][bank[,4] == "RURAL"] = 3

bank[,4][bank[,4] == "SUBURBAN"] = 4

as.numeric(bank[,4])

bank[,6][bank[,6] == "YES"] = 1

bank[,6][bank[,6] == "NO"] = 2

as.numeric(bank[,6])

bank[,8][bank[,8] == "YES"] = 1

bank[,8][bank[,8] == "NO"] = 2

as.numeric(bank[,8])

bank[,9][bank[,9] == "YES"] = 1

bank[,9][bank[,9] == "NO"] = 2

as.numeric(bank[,9])

bank[,10][bank[,10] == "YES"] = 1

bank[,10][bank[,10] == "NO"] = 2

as.numeric(bank[,10])

bank[,11][bank[,11] == "YES"] = 1

bank[,11][bank[,11] == "NO"] = 2

as.numeric(bank[,11])

bank[,12][bank[,12] == "YES"] = 1

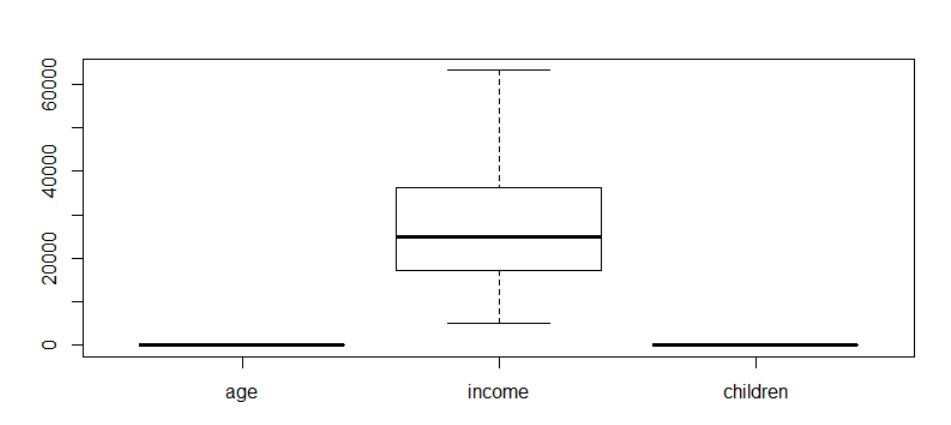
bank[,12][bank[,12] == "NO"] = 2

as.numeric(bank[,12])

> # First check the complete set of components for outliers

> mydataNUM<-bank[,c(2,5,7)]

> mydata<- bank[,2:12]



step 4 ： checking the outliers

> # since there is not any outliers, so we do not do additional replace

step 5 ： Choosing the data mining task

> ## Calculating variance and storing at the first index in wss

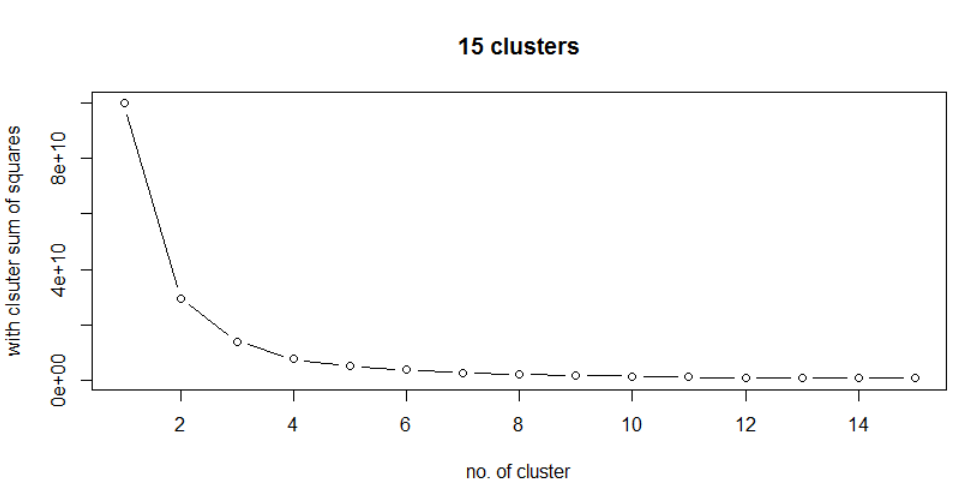
> ## given columns to caculate the variance and sum them,

> wss <-(nrow(mydata)-1)\*sum(apply(mydata,2,var))

> wss

[1] 99671499206

|  |
| --- |
| Step 5 : use k-means algorithm for this clustering  > ## iterate through wss array 15 times and sum up all the variance in  every iteration and store it in wss array  > for(i in 2:15)wss[i]<- sum(fit=kmeans(mydata,centers=i,15)$withinss)  > ## plot each iteration to display the elbow graph  > plot(1:9,wss,type="b",main="9 clusters",xlab="no. of cluster",ylab="with  clsuter sum of squares") |
|  |
| |  | | --- | | > | |



Step 6 : Searching for patterns of interest in a particular representational form

> ##As we can see from the above output the slope of the graph changes majorly in 2 iteration, hence we consider the optimized number of cluster as 2.

But follow the example form WEKA, we set the number of cluster as 6.

> fit <- kmeans(mydata,6)

Step 7 : Let’s check the summary of the kmeans objects

> fit

K-means clustering with 6 clusters of sizes 83, 139, 130, 45, 142, 61

Cluster means:

age sex region income married children car save\_act current\_act mortgage

1 50.09639 1.433735 1.855422 36048.88 1.397590 1.1445783 1.421687 1.337349 1.228916 1.662651

2 37.30216 1.489209 1.892086 20587.08 1.273381 0.9856115 1.568345 1.438849 1.280576 1.633094

3 45.26154 1.584615 1.992308 27755.18 1.369231 0.9230769 1.430769 1.353846 1.269231 1.615385

4 62.64444 1.600000 2.155556 55105.82 1.444444 1.1777778 1.488889 1.000000 1.200000 1.688889

5 27.81690 1.471831 1.802817 12895.36 1.359155 0.9929577 1.577465 1.359155 1.218310 1.676056

6 56.40984 1.426230 2.000000 44945.61 1.229508 1.0000000 1.491803 1.000000 1.196721 1.672131

pep

1 1.433735

2 1.539568

3 1.561538

4 1.177778

5 1.697183

6 1.573770

Clustering vector:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

2 3 5 2 4 1 5 3 3 3 4 3 5 4 2 2 2 6 3 2 4 5 1 5 5

26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

6 2 2 3 3 2 5 2 3 1 2 5 5 2 3 3 3 4 3 5 4 5 5 1 2

51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75

5 2 5 3 4 6 3 2 5 3 3 1 3 5 5 2 5 1 2 5 5 5 1 2 4

76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

1 3 6 5 1 2 3 5 5 2 3 5 2 5 3 3 3 1 6 6 5 5 3 5 5

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125

3 5 5 1 1 1 6 5 5 2 1 3 1 1 6 6 2 5 2 6 3 4 2 1 6

126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150

1 3 6 5 2 2 6 5 1 4 3 5 1 6 4 1 2 2 6 2 4 2 2 3 3

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

3 1 4 3 2 5 3 2 2 5 2 5 2 5 3 2 5 5 3 1 6 2 3 3 6

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

4 1 3 5 1 2 4 1 2 3 1 2 1 1 2 4 5 6 5 1 2 2 6 5 2

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

3 6 2 2 5 5 4 5 1 2 3 1 5 5 5 2 5 3 3 3 3 6 1 6 6

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

3 4 3 3 5 3 3 5 3 6 5 4 2 2 5 5 4 1 2 3 1 5 2 2 2

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

4 1 5 2 5 3 3 2 2 5 2 3 2 6 2 2 1 3 1 5 4 4 5 3 3

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

5 3 2 3 1 2 5 4 1 5 5 3 3 5 1 5 1 6 2 2 1 3 5 2 5

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

5 5 3 6 4 5 4 3 3 5 3 1 3 5 2 6 2 1 2 2 1 2 5 2 3

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

5 1 2 4 2 4 5 3 2 2 3 3 1 1 2 3 6 4 2 2 6 1 6 1 2

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

3 3 3 6 3 4 2 5 1 2 4 2 3 5 4 2 3 6 5 1 6 1 5 3 6

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

2 1 3 2 5 3 1 5 1 3 1 5 3 5 5 6 2 1 2 3 2 3 1 4 1

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

2 2 1 2 5 1 6 2 2 5 2 5 4 5 6 2 5 2 3 2 6 6 5 6 2

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

3 3 1 2 3 3 4 2 3 2 5 5 3 5 5 3 6 1 1 5 3 6 3 6 1

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

3 1 3 4 5 5 5 4 5 6 2 3 2 5 2 3 2 5 5 1 6 3 2 3 2

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

5 3 3 2 6 5 6 1 5 2 6 2 5 5 2 3 2 3 6 6 1 6 6 6 2

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

3 1 3 5 4 3 2 5 4 5 5 5 6 5 1 5 2 1 2 3 5 2 5 5 2

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

2 5 5 3 6 4 3 3 3 5 5 1 4 1 5 2 1 5 6 2 1 5 5 2 4

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

5 1 6 1 1 5 2 3 3 3 5 2 4 2 2 3 2 2 3 2 1 1 3 2 3

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

3 2 6 5 3 4 5 2 3 2 5 2 3 6 5 2 3 2 4 3 6 5 5 5 3

Within cluster sum of squares by cluster:

[1] 418240742 635515801 653344244 544888855 1053251143 502615169

(between\_SS / total\_SS = 96.2 %)

Available components:

[1] "cluster" "centers" "totss" "withinss" "tot.withinss" "betweenss"

[7] "size" "iter" "ifault"

Step 7 : checking withinss i.e. the intra cluster bond strength factor for each cluster

> ## checking withinss i.e. the intra cluster bond strength factor for each cluster

> fit$withinss

[1] 418240742 635515801 653344244 544888855 1053251143 502615169

> ## checking betweenss i.e. the inter cluster distance between cluster

> fit$betweenss

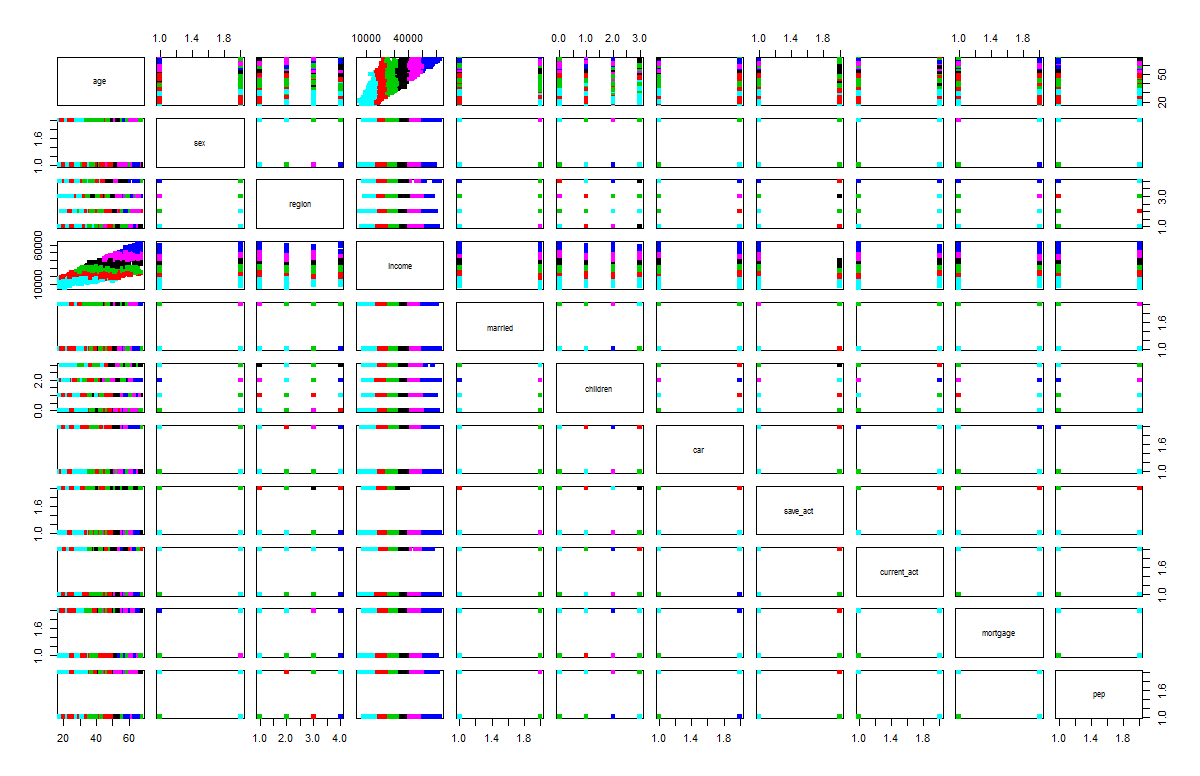
[1] 95863643252

> fit$size

[1] 83 139 130 45 142 61

> ####Interpreting mined patterns

> plot(mydata,col=fit$cluster,pch=15)



> fit$centers

age sex region income married children car save\_act current\_act mortgage

1 50.09639 1.433735 1.855422 36048.88 1.397590 1.1445783 1.421687 1.337349 1.228916 1.662651

2 37.30216 1.489209 1.892086 20587.08 1.273381 0.9856115 1.568345 1.438849 1.280576 1.633094

3 45.26154 1.584615 1.992308 27755.18 1.369231 0.9230769 1.430769 1.353846 1.269231 1.615385

4 62.64444 1.600000 2.155556 55105.82 1.444444 1.1777778 1.488889 1.000000 1.200000 1.688889

5 27.81690 1.471831 1.802817 12895.36 1.359155 0.9929577 1.577465 1.359155 1.218310 1.676056

6 56.40984 1.426230 2.000000 44945.61 1.229508 1.0000000 1.491803 1.000000 1.196721 1.672131

pep

1 1.433735

2 1.539568

3 1.561538

4 1.177778

5 1.697183

6 1.573770

**Step 8 ： do the plotcluster**

> library(cluster)

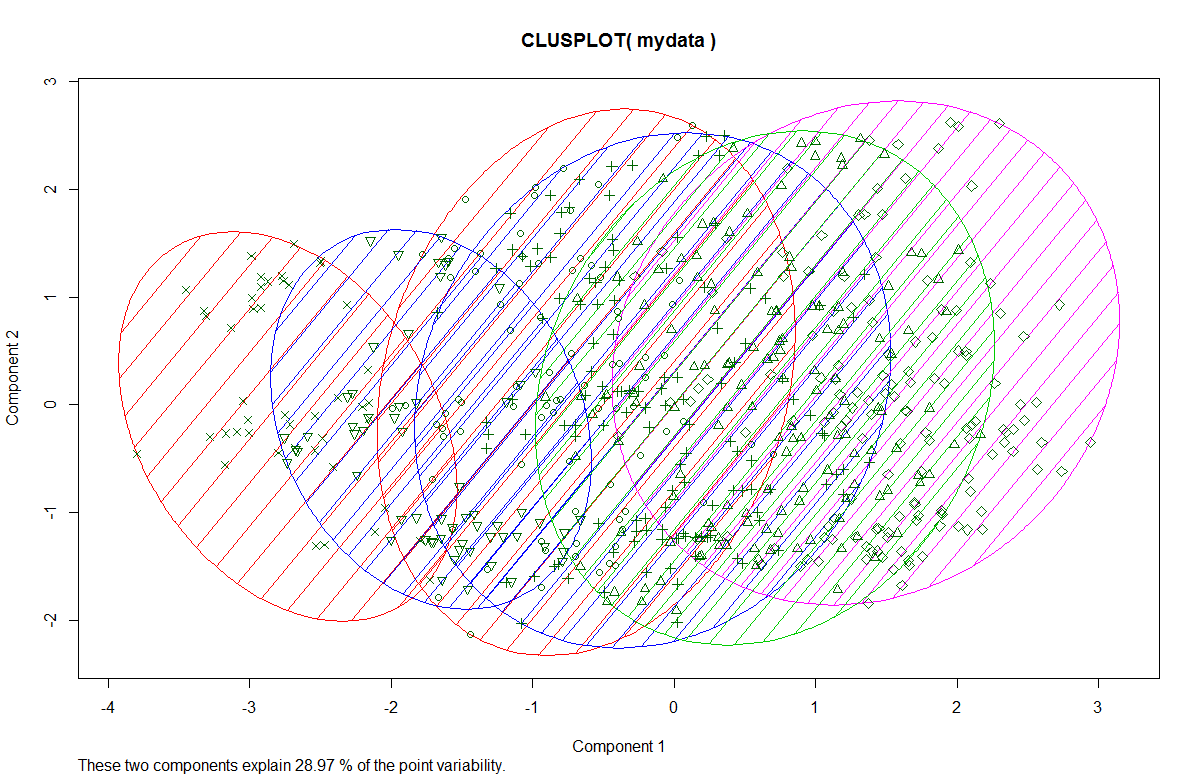
> #install.packages("fpc")

> library(fpc)

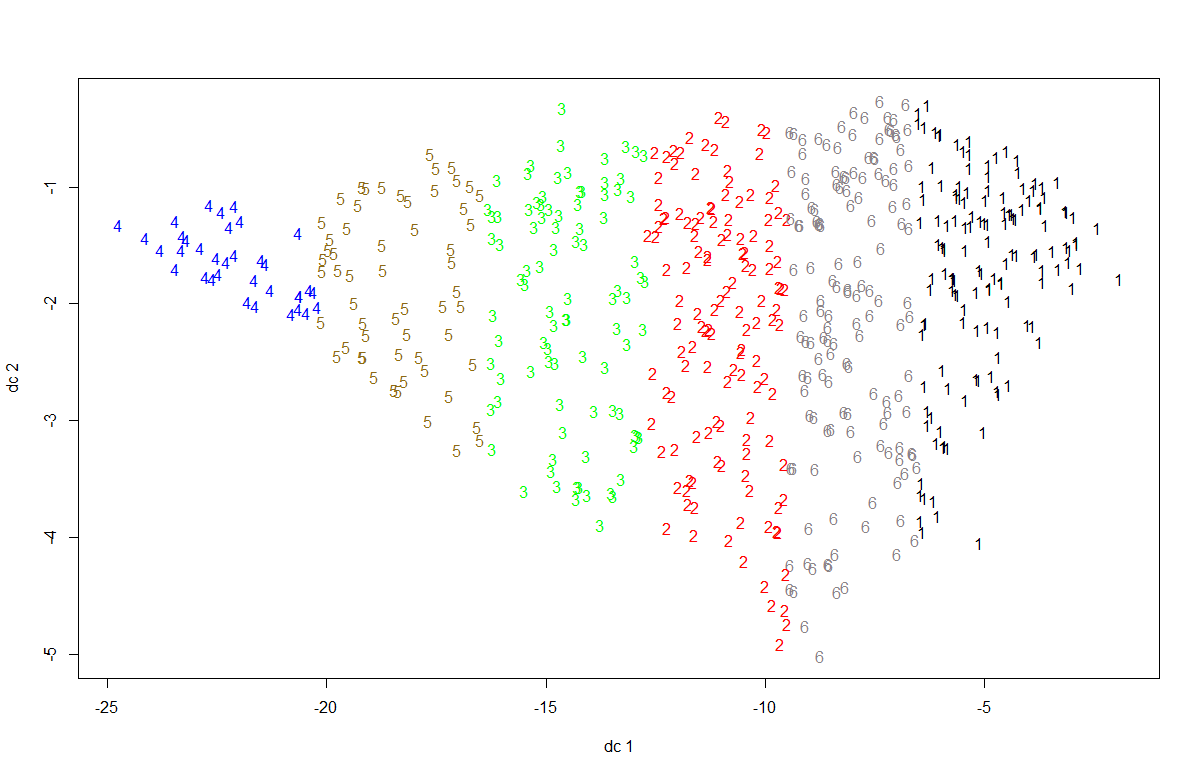
> #plotcluster(mydata,fit$cluster)

> #points(fit$centers,col=1:8,pch=16)

> clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE, labels=0, lines=0)



> plotcluster(mydata,fit$cluster)



>clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE, labels=0, lines=0)

Analysis report:

1. normolize the dataset, remove the NA.
2. convert the factor to number for k-means implement
3. check and deal with outliers.
4. choose the number of clusters we need
5. check the clusters, size, means, vectors, sum of squares
6. plot the correlation
7. check centers, clusters

Actually, in this case, we use K-means algorithm to classify group. k-means is sensitive to outliers, but in this case, do not have outliers, it is a good way to do data mining.The k value here as I mentioned before, it is supposed to be 2 according to the graph which plotted 9 iterations.

Every group has similar features, as the result shows, people who is 62 years old, their income higher, and the number of the children they have also a little bit more than other generation. The 42 age generation they have less children. The features are below with size.

As we can see, For the business men, they can make decision depend on size of group and the all kinds of features that this group have and make objective strategy more effectively.

Grou1:83

Group2:139

Group3:130

Group4:45

Group5:142

Group6:61

Age sex region income married children car save\_act

1 50.09639 1.433735 1.855422 36048.88 1.397590 1.1445783 1.421687 1.337349

2 37.30216 1.489209 1.892086 20587.08 1.273381 0.9856115 1.568345 1.438849

3 45.26154 1.584615 1.992308 27755.18 1.369231 0.9230769 1.430769 1.353846

4 62.64444 1.600000 2.155556 55105.82 1.444444 1.1777778 1.488889 1.000000

5 27.81690 1.471831 1.802817 12895.36 1.359155 0.9929577 1.577465 1.359155

6 56.40984 1.426230 2.000000 44945.61 1.229508 1.0000000 1.491803 1.000000

current\_act mortgage pep

1 1.228916 1.662651 1.433735

2 1.280576 1.633094 1.539568

3 1.269231 1.615385 1.561538

4 1.200000 1.688889 1.177778

5 1.218310 1.676056 1.697183

6 1.196721 1.672131 1.573770