King Saud University
College of Computer and Information Sciences
Information Technology department

IT 326: Data Mining

Course Project

Body signal of smoking

Project Report

Group #:	Team5	
Section:	52846	
Group members:	Name	ID
	Sarah k Jwuied	442201381
	Nouf Saleh Aldakheel	442202526
	Basma Alamoud	442200304

2/4/2023

1 Problem

Smoking is a complex behavior with a heritability as high as 50% and it is a leading cause of preventable disease, disability, and death in the world. By this dataset we chose, that is a collection of basic health biological signal data.it provides us an opportunity to predict if the person smokes or not based on their body signal, which can help the hospitals and other benefits.

2 Data Mining Task

The used data mining task is classification. The class attribute is Smoke. This class attribute will determine from the body signal attributes if the person smokes or not. The goal of this data mining task is to determine the presence or absence of smoking through bio-signals, and the other data mining task is clustering. We are going to analyze the attribute data by partitioning the data into groups where the similar data are classed together, which can help us manage our data.

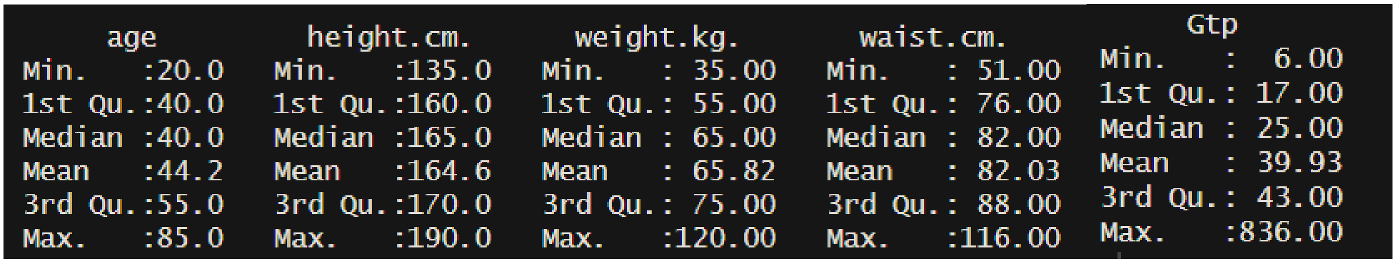
3 Data

-This data set contains: 26 attributes and 5999 Observations.
-Source: Body signal of smoking | Kaggle

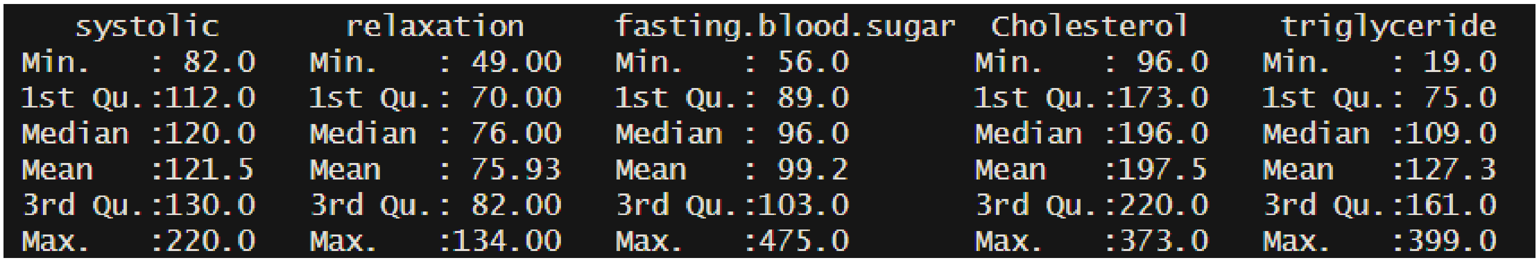
Attributs	Type
gender	binary
age	numeric
height	numeric
weight	numeric
waist	numeric
eyesight(left)	ordinal
eyesight(right)	ordinal
hearing(left)	binary
hearing(right)	binary
systolic	numeric

relaxation	numeric
fasting blood sugar	numeric
Cholesterol	numeric
triglyceride	numeric
HDL	numeric
LDL	numeric
hemoglobin	numeric
Urine protein	numeric
serum creatinine	numeric
AST	numeric
ALT	numeric
Gtp	numeric
oral	binary
dental caries	binary
tartar	binary
smoke	binary

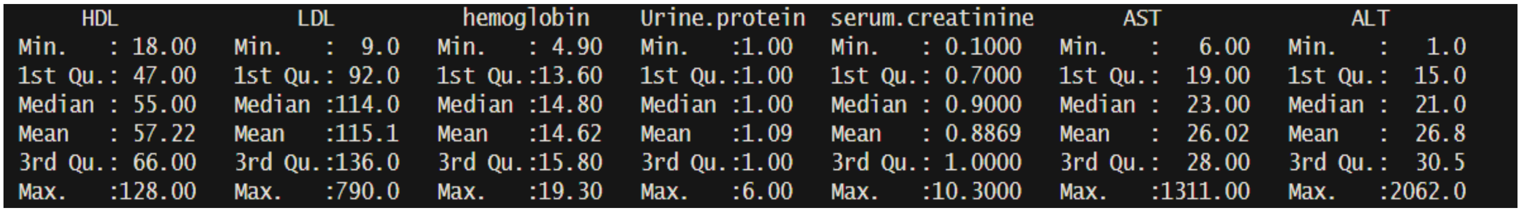
- **Five number summary**



```
      age          height.cm.      weight.kg.        waist.cm.          Gtp
 Min.   :20.0   Min.   :135.0   Min.   : 35.00   Min.   : 51.00   Min.   :  6.00
 1st Qu.:40.0   1st Qu.:160.0   1st Qu.: 55.00   1st Qu.: 76.00   1st Qu.: 17.00
 Median :40.0   Median :165.0   Median : 65.00   Median : 82.00   Median : 25.00
 Mean   :44.2   Mean   :164.6   Mean   : 65.82   Mean   : 82.03   Mean   : 39.93
 3rd Qu.:55.0   3rd Qu.:170.0   3rd Qu.: 75.00   3rd Qu.: 88.00   3rd Qu.: 43.00
 Max.   :85.0   Max.   :190.0   Max.   :120.00   Max.   :116.00   Max.   :836.00
```



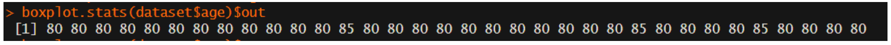
```
    systolic        relaxation     fasting.blood.sugar  Cholesterol     triglyceride
 Min.   : 82.0   Min.   : 49.00   Min.   : 56.0       Min.   : 96.0   Min.   : 19.0
 1st Qu.:112.0   1st Qu.: 70.00   1st Qu.: 89.0       1st Qu.:173.0   1st Qu.: 75.0
 Median :120.0   Median : 76.00   Median : 96.0       Median :196.0   Median :109.0
 Mean   :121.5   Mean   : 75.93   Mean   : 99.2       Mean   :197.5   Mean   :127.3
 3rd Qu.:130.0   3rd Qu.: 82.00   3rd Qu.:103.0       3rd Qu.:220.0   3rd Qu.:161.0
 Max.   :220.0   Max.   :134.00   Max.   :475.0       Max.   :373.0   Max.   :399.0
```

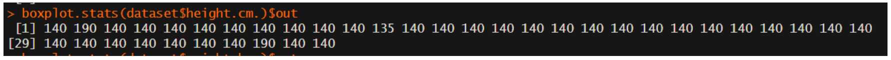


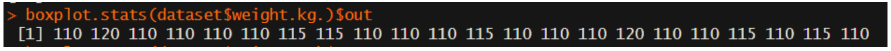
```
      HDL              LDL           hemoglobin    Urine.protein  serum.creatinine       AST               ALT        
 Min.   : 18.00   Min.   :  9.0   Min.   : 4.90   Min.   :1.00   Min.   : 0.1000   Min.   :   6.00   Min.   :   1.0  
 1st Qu.: 47.00   1st Qu.: 92.0   1st Qu.:13.60   1st Qu.:1.00   1st Qu.: 0.7000   1st Qu.:  19.00   1st Qu.:  15.0  
 Median : 55.00   Median :114.0   Median :14.80   Median :1.00   Median : 0.9000   Median :  23.00   Median :  21.0  
 Mean   : 57.22   Mean   :115.1   Mean   :14.62   Mean   :1.09   Mean   : 0.8869   Mean   :  26.02   Mean   :  26.8  
 3rd Qu.: 66.00   3rd Qu.:136.0   3rd Qu.:15.80   3rd Qu.:1.00   3rd Qu.: 1.0000   3rd Qu.:  28.00   3rd Qu.:  30.5  
 Max.   :128.00   Max.   :790.0   Max.   :19.30   Max.   :6.00   Max.   :10.3000   Max.   :1311.00   Max.   :2062.0  
```

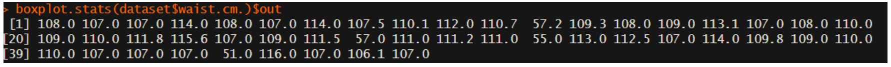
- outliers

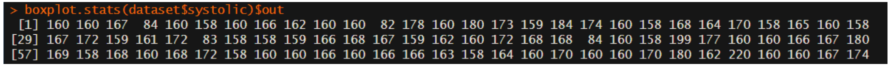
Attributs	Number of outliers

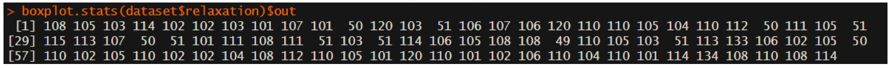
age	[1:34]
	
	`> boxplot.stats(dataset$age)$out`
	`[1] 80 80 80 80 80 80 80 80 80 80 80 80 85 80 80 80 80 80 80 80 80 80 80 80 85 80 80 80 80 85 80 80 80 80`

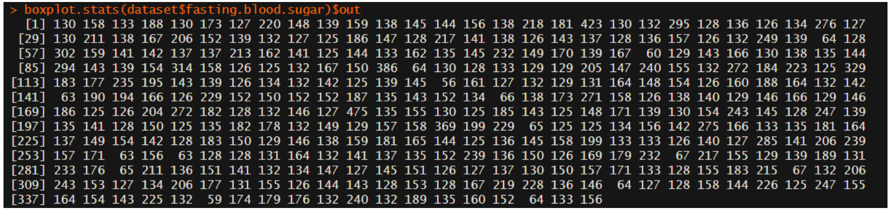
height	[1:38]
	
	`> boxplot.stats(dataset$height.cm.)$out`
	` [1] 140 190 140 140 140 140 140 140 140 140 140 135 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140 140`
	`[29] 140 140 140 140 140 140 140 190 140 140`

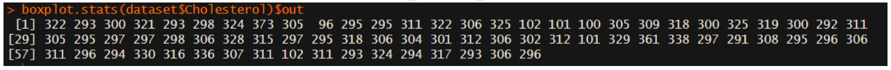
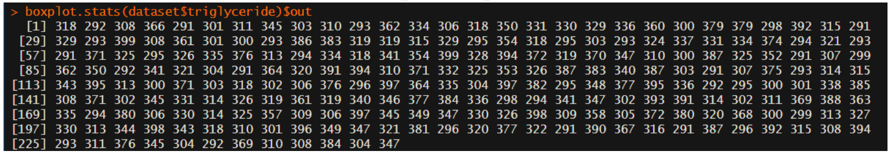
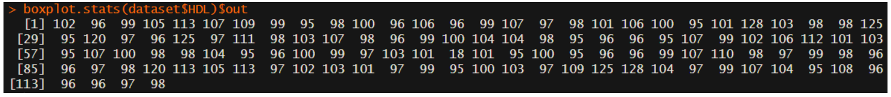
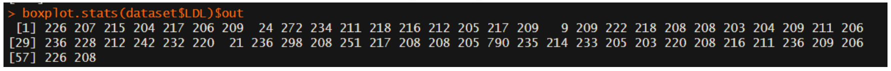
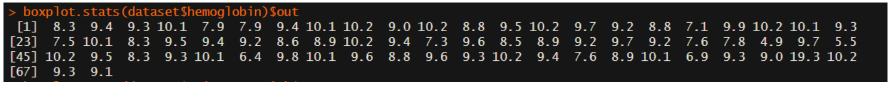
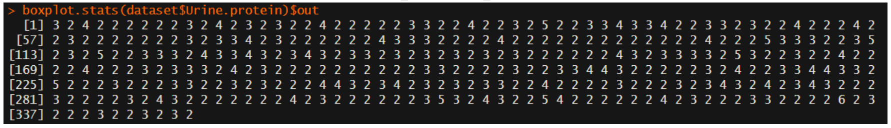
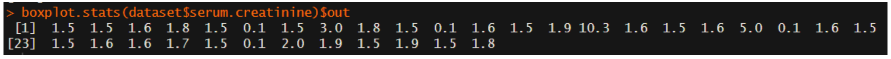
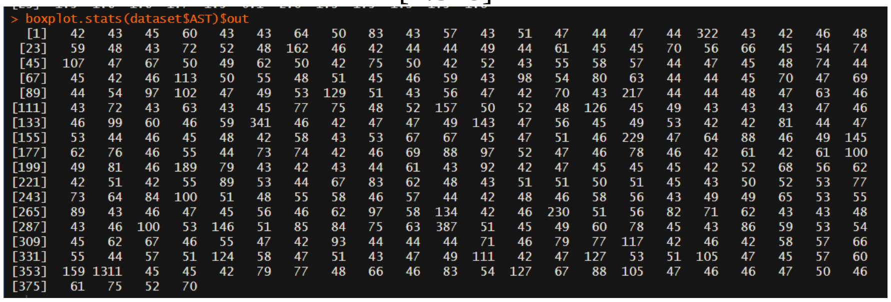
weight	[1:22]
	
	`> boxplot.stats(dataset$weight.kg.)$out`
	`[1] 110 120 110 110 110 110 115 115 110 110 110 115 110 110 110 120 110 110 115 110 115 110`

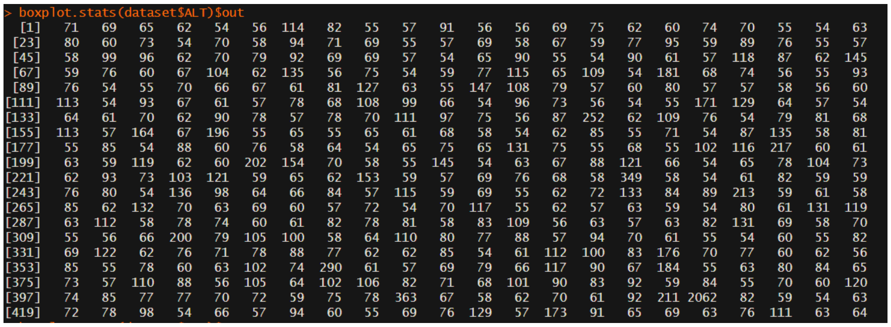
waist	[1:47]
	
	`> boxplot.stats(dataset$waist.cm.)$out`
	` [1] 108.0 107.0 107.0 114.0 108.0 107.0 114.0 107.5 110.1 112.0 110.7  57.2 109.3 108.0 109.0 113.1 107.0 108.0 110.0`
	`[20] 109.0 110.0 111.8 115.6 107.0 109.0 111.5  57.0 111.0 111.2 111.0  55.0 113.0 112.5 107.0 114.0 109.8 109.0 110.0`
	`[39] 110.0 107.0 107.0 107.0  51.0 116.0 107.0 106.1 107.0`

systolic	[1:84]
	
	`> boxplot.stats(dataset$systolic)$out`
	` [1] 160 160 167  84 160 158 160 166 162 160 160  82 178 160 180 173 159 184 174 160 158 168 164 170 158 165 160 158`
	`[29] 167 172 159 161 172  83 158 158 159 166 168 167 159 162 160 172 168 168  84 160 158 199 177 160 160 166 167 180`
	`[57] 169 158 168 160 168 172 158 160 160 166 160 166 166 163 158 164 160 170 160 160 170 180 162 220 160 160 167 174`

relaxation	[1:83]
	
	`> boxplot.stats(dataset$relaxation)$out`
	` [1] 108 105 103 114 102 102 103 101 107 101  50 120 103  51 106 107 106 120 110 110 105 104 110 112  50 111 105  51`
	`[29] 115 113 107  50  51 101 111 108 111  51 103  51 114 106 105 108 108  49 110 105 103  51 113 133 106 102 105  50`
	`[57] 110 102 105 110 102 102 104 108 112 110 105 101 120 110 101 102 106 110 104 110 101 114 134 108 110 108 114`

fasting blood sugar	[1:355]
	
	`> boxplot.stats(dataset$fasting.blood.sugar)$out`
	`  [1] 130 158 133 188 130 173 127 220 148 139 159 138 145 144 156 138 218 181 423 130 132 295 128 136 126 134 276 127`
	` [29] 130 211 138 167 206 152 139 132 127 125 186 147 128 217 141 138 126 143 137 128 136 157 126 132 249 139  64 128`
	` [57] 302 159 141 142 137 137 213 162 141 125 144 133 162 135 145 232 149 170 139 167  60 129 143 166 130 138 135 144`
	` [85] 294 143 139 154 314 158 126 125 132 167 150 386  64 130 128 133 129 129 205 147 240 155 132 272 184 223 125 329`
	`[113] 183 177 235 195 143 139 126 134 132 142 125 139 145  56 161 127 132 129 131 164 148 154 126 160 188 164 132 142`
	`[141]  63 190 194 166 126 229 152 150 152 152 187 135 143 152 134  66 138 173 271 158 126 138 140 129 146 166 129 146`
	`[169] 186 125 126 204 272 182 128 132 146 127 475 135 155 130 125 185 143 125 148 171 139 130 154 243 145 128 247 139`
	`[197] 135 141 128 150 125 135 182 178 132 149 129 157 158 369 199 229  65 125 125 134 156 142 275 166 133 135 181 164`
	`[225] 137 149 154 142 128 183 150 129 146 138 159 181 165 144 125 136 145 158 199 133 133 126 140 127 285 141 206 239`
	`[253] 157 171  63 156  63 128 128 131 164 132 141 137 135 152 239 136 150 126 169 179 232  67 217 155 129 139 189 131`
	`[281] 233 176  65 211 136 151 141 132 134 147 127 145 151 126 127 137 130 150 157 171 133 128 155 183 215  67 132 206`
	`[309] 243 153 127 134 206 177 131 155 126 144 143 128 153 128 167 219 228 136 146  64 127 128 158 144 226 125 247 155`
	`[337] 164 154 143 225 132  59 174 179 176 132 240 132 189 135 160 152  64 133 156`

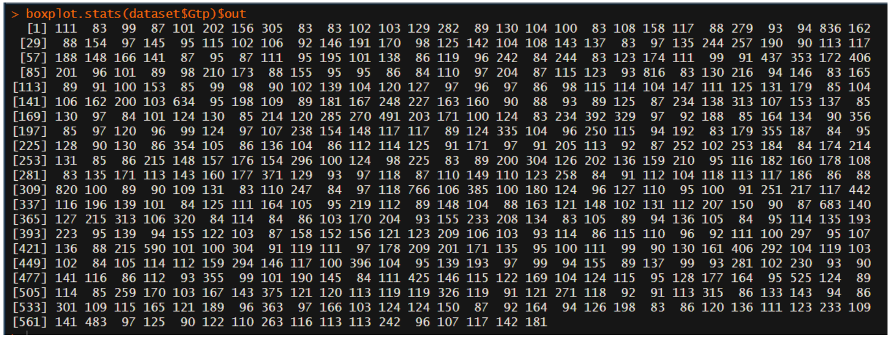
Cholesterol	[1:73]  > boxplot.stats(dataset$Cholesterol)$out [1] 322 293 300 321 293 298 324 373 305 96 295 295 311 322 306 325 102 101 100 305 309 318 300 325 319 300 292 311 [29] 305 295 297 297 298 306 328 315 297 295 318 306 304 301 312 306 302 312 101 329 361 338 297 291 308 295 296 306 [57] 311 296 294 330 316 336 307 311 102 311 293 324 294 317 293 306 296
triglyceride	[1:236]  > boxplot.stats(dataset$triglyceride)$out [1] 318 292 308 366 291 301 311 345 303 310 293 362 334 306 318 350 331 330 329 336 360 300 379 379 298 392 315 291 [29] 329 293 399 308 361 301 300 293 386 383 319 319 315 329 295 354 318 295 303 293 324 337 331 334 374 294 321 293 [57] 291 371 325 295 326 335 376 313 294 334 318 341 354 399 328 394 372 319 370 347 310 300 387 325 352 291 307 299 [85] 362 350 292 341 321 304 291 364 320 391 394 310 371 332 325 353 326 387 383 340 387 303 291 307 375 293 314 315 [113] 343 395 313 300 371 303 318 302 306 376 296 397 364 335 304 397 382 295 348 377 395 336 292 295 300 301 338 385 [141] 308 371 302 345 331 314 326 319 361 319 340 346 377 384 336 298 294 341 347 302 393 391 314 302 311 369 388 363 [169] 335 294 380 306 330 314 325 357 309 306 397 345 349 347 330 326 398 309 358 305 372 380 320 368 300 299 313 327 [197] 330 313 344 398 343 318 310 301 396 349 347 321 381 296 320 377 322 291 390 367 316 291 387 296 392 315 308 394 [225] 293 311 376 345 304 292 369 310 308 384 304 347
HDL	[1:116]  > boxplot.stats(dataset$HDL)$out [1] 102 96 99 105 113 107 109 99 95 98 100 96 106 96 99 107 97 98 101 106 100 95 101 128 103 98 98 125 [29] 95 120 97 96 125 97 111 98 103 107 98 96 99 100 104 104 98 95 96 96 95 107 99 102 106 112 101 103 [57] 95 107 100 98 98 104 95 96 100 99 97 103 101 18 101 95 100 95 96 96 99 107 110 98 97 99 98 96 [85] 96 97 98 120 113 105 113 97 102 103 101 97 99 95 100 103 97 109 125 128 104 97 99 107 104 95 108 96 [113] 96 96 97 98
LDL	[1:58]  > boxplot.stats(dataset$LDL)$out [1] 226 207 215 204 217 206 209 24 272 234 211 218 216 212 205 217 209 9 209 222 218 208 208 203 204 209 211 206 [29] 236 228 212 242 232 220 21 236 298 208 251 217 208 208 205 790 235 214 233 205 203 220 208 216 211 236 209 206 [57] 226 208
hemoglobin	[1:68]  > boxplot.stats(dataset$hemoglobin)$out [1] 8.3 9.4 9.3 10.1 7.9 7.9 9.4 10.1 10.2 9.0 10.2 8.8 9.5 10.2 9.7 9.2 8.8 7.1 9.9 10.2 10.1 9.3 [23] 7.5 10.1 8.3 9.5 9.4 9.2 8.6 8.9 10.2 9.4 7.3 9.6 8.5 8.9 9.2 9.7 9.2 7.6 7.8 4.9 9.7 5.5 [45] 10.2 9.5 8.3 9.3 10.1 6.4 9.8 10.1 9.6 8.8 9.6 9.3 10.2 9.4 7.6 8.9 10.1 6.9 9.3 9.0 19.3 10.2 [67] 9.3 9.1
Urine protein	[1:346]  > boxplot.stats(dataset$Urine.protein)$out [1] 3 2 4 2 2 2 2 2 2 3 2 4 2 3 2 3 2 2 4 2 2 2 2 2 3 3 2 2 4 2 2 3 2 5 2 2 3 3 4 3 3 4 2 2 3 3 2 3 2 2 4 2 2 2 4 2 [57] 2 3 2 2 2 2 2 2 2 3 2 3 3 4 2 3 2 2 2 2 2 2 4 3 3 3 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 5 3 3 3 2 2 3 5 [113] 2 3 2 5 2 2 3 3 3 2 4 3 3 4 3 2 3 4 3 2 3 3 2 3 2 3 2 3 2 3 2 3 2 2 2 2 2 2 4 3 2 3 3 3 3 2 5 3 2 2 3 2 2 4 2 2 [169] 2 2 4 2 2 2 3 2 3 3 3 2 4 2 3 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 2 3 2 2 3 3 4 4 3 2 2 2 2 2 3 2 4 2 2 3 3 4 4 3 3 2 [225] 5 2 2 2 3 2 2 2 3 3 2 2 3 2 3 2 2 2 4 4 3 2 3 4 2 3 2 3 2 3 3 2 2 4 2 2 2 2 3 2 2 2 3 2 3 4 3 2 4 2 3 4 3 2 2 2 [281] 3 2 2 2 2 3 2 4 3 2 2 2 2 2 2 2 4 2 3 2 2 2 2 2 2 3 5 3 2 4 3 2 2 5 4 2 2 2 2 2 2 4 2 3 2 2 2 3 3 2 2 2 2 6 2 3 [337] 2 2 2 3 2 2 3 2 3 2
serum creatinine	[1:34]  > boxplot.stats(dataset$serum.creatinine)$out [1] 1.5 1.5 1.6 1.8 1.5 0.1 1.5 3.0 1.8 1.5 0.1 1.6 1.5 1.9 10.3 1.6 1.5 1.6 5.0 0.1 1.6 1.5 [23] 1.5 1.6 1.6 1.7 1.5 0.1 2.0 1.9 1.5 1.9 1.5 1.8
AST	[1:378]  > boxplot.stats(dataset$AST)$out [1] 42 43 45 60 43 43 64 50 83 43 57 43 51 47 44 47 44 322 43 42 46 48 [23] 59 48 43 72 52 48 162 46 42 44 44 49 44 61 45 45 70 56 66 45 54 74 [45] 107 47 67 50 49 62 50 42 75 50 42 52 43 55 58 57 44 47 45 48 74 44 [67] 45 42 46 113 50 55 48 51 45 46 59 43 98 54 80 63 44 44 45 70 47 69 [89] 44 54 97 102 47 49 53 129 51 43 56 47 42 70 43 217 44 44 48 47 63 46 [111] 43 72 43 63 43 45 77 75 48 52 157 50 52 48 126 45 49 43 43 43 47 46 [133] 46 99 60 46 59 341 46 42 47 47 49 143 47 56 45 49 53 42 42 81 44 47 [155] 53 44 46 45 48 42 58 43 53 67 67 45 47 51 46 229 47 64 88 46 49 145 [177] 62 76 46 55 44 73 74 42 46 69 88 97 52 47 46 78 46 42 61 42 61 100 [199] 49 81 46 189 79 43 42 43 44 61 43 92 42 47 45 45 45 42 52 68 56 62 [221] 42 51 42 55 89 53 44 67 83 62 48 43 51 51 50 51 45 43 50 52 53 77 [243] 73 64 84 100 51 48 55 58 46 57 44 42 48 46 58 56 43 49 49 65 53 55 [265] 89 43 46 47 45 56 46 62 97 58 134 42 46 230 51 56 82 71 62 43 43 48 [287] 43 46 100 53 146 51 85 84 75 63 387 51 45 49 60 78 45 43 86 59 53 54 [309] 45 62 67 46 55 47 42 93 44 44 44 71 46 79 77 117 42 46 42 58 57 66 [331] 55 44 57 51 124 58 47 51 43 47 49 111 42 47 127 53 51 105 47 45 57 60 [353] 159 1311 45 45 42 79 77 48 66 46 83 54 127 67 88 105 47 46 46 47 50 46 [375] 61 75 52 70
ALT	[1:440]



```
> boxplot.stats(dataset$ALT)$out
  [1]    71    69    65    62    54    56   114    82    55    57    91    56    56    69    75    62    60    74    70    55    54    63
 [23]    80    60    73    54    70    58    94    71    69    55    57    69    58    67    59    77    95    59    89    76    55    57
 [45]    58    99    96    62    70    79    92    69    69    57    54    65    90    55    54    90    61    57   118    87    62   145
 [67]    59    76    60    67   104    62   135    56    75    54    59    77   115    65   109    54   181    68    74    56    55    93
 [89]    76    54    55    70    66    67    61    81   127    63    55   147   108    79    57    60    80    57    57    58    56    60
[111]   113    54    93    67    61    57    78    68   108    99    66    54    96    73    56    54    55   171   129    64    57    54
[133]    64    61    70    62    90    78    57    78    70   111    97    75    56    87   252    62   109    76    54    79    81    68
[155]   113    57   164    67   196    55    65    55    65    61    68    58    54    62    85    55    71    54    87   135    58    81
[177]    55    85    54    88    60    76    58    64    54    65    75    65   131    75    55    68    55   102   116   217    60    61
[199]    63    59   119    62    60   202   154    70    58    55   145    54    63    67    88   121    66    54    65    78   104    73
[221]    62    93    73   103   121    59    65    62   153    59    57    69    76    68    58   349    58    54    61    82    59    59
[243]    76    80    54   136    98    64    66    84    57   115    59    69    55    62    72   133    84    89   213    59    61    58
[265]    85    62   132    70    63    69    60    57    72    54    70   117    55    62    57    63    59    54    80    61   131   119
[287]    63   112    58    78    74    60    61    82    78    81    58    83   109    56    63    57    63    82   131    69    58    70
[309]    55    56    66   200    79   105   100    58    64   110    80    77    88    57    94    70    61    55    54    60    55    82
[331]    69   122    62    76    71    78    88    77    62    62    85    54    61   112   100    83   176    70    77    60    62    56
[353]    85    55    78    60    63   102    74   290    61    57    69    79    66   117    90    67   184    55    63    80    84    65
[375]    73    57   110    88    56   105    64   102   106    82    71    68   101    90    83    92    59    84    55    70    60   120
[397]    74    85    77    77    70    72    59    75    78   363    67    58    62    70    61    92   211  2062    82    59    54    63
[419]    72    78    98    54    66    57    94    60    55    69    76   129    57   173    91    65    69    63    76   111    63    64
```

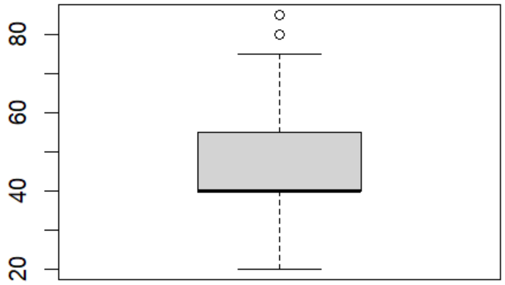
Gtp

[1:577]

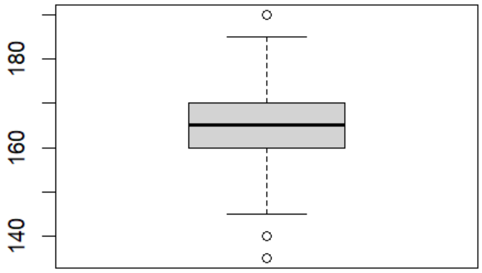


```
> boxplot.stats(dataset$Gtp)$out
  [1] 111  83  99  87 101 202 156 305  83  83 102 103 129 282  89 130 104 100  83 108 158 117  88 279  93  94 836 162
 [29]  88 154  97 145  95 115 102 106  92 146 191 170  98 125 142 104 108 143 137  83  97 135 244 257 190  90 113 117
 [57] 188 148 166 141  87  95  87 111  95 195 101 138  86 119  96 242  84 244  83 123 174 111  99  91 437 353 172 406
 [85] 201  96 101  89  98 210 173  88 155  95  95  86  84 110  97 204  87 115 123  93 816  83 130 216  94 146  83 165
[113]  89  91 100 153  85  99  98  90 102 139 104 120 127  97  96  97  86  98 115 114 104 147 111 125 131 179  85 104
[141] 106 162 200 103 634  95 198 109  89 181 167 248 227 163 160  90  88  93  89 125  87 234 138 313 107 153 137  85
[169] 130  97  84 101 124 130  85 214 120 285 270 491 203 171 100 124  83 234 392 329  97  92 188  85 164 134  90 356
[197]  85  97 120  96  99 124  97 107 238 154 148 117 117  89 124 335 104  96 250 115  94 192  83 179 355 187  84  95
[225] 128  90 130  86 354 105  86 136 104  86 112 114 125  91 171  97  91 205 113  92  87 252 102 253 184  84 174 214
[253] 131  85  86 215 148 157 176 154 296 100 124  98 225  83  89 200 304 126 202 136 159 210  95 116 182 160 178 108
[281]  83 135 171 113 143 160 177 371 129  93  97 118  87 110 149 110 123 258  84  91 112 104 118 113 117 186  86  88
[309] 820 100  89  90 109 131  83 110 247  84  97 118 766 106 385 100 180 124  96 127 110  95 100  91 251 217 117 442
[337] 116 196 139 101  84 125 111 164 105  95 219 112  89 148 104  88 163 121 148 102 131 112 207 150  90  87 683 140
[365] 127 215 313 106 320  84 114  84  86 103 170 204  93 155 233 208 134  83 105  89  94 136 105  84  95 114 135 193
[393] 223  95 139  94 155 122 103  87 158 152 156 121 123 209 106 103  93 114  86 115 110  96  92 111 100 297  95 107
[421] 136  88 215 590 101 100 304  91 119 111  97 178 209 201 171 135  95 100 111  99  90 130 161 406 292 104 119 103
[449] 102  84 105 114 112 159 294 146 117 100 396 104  95 139 193  97  99  94 155  89 137  99  93 281 102 230  93  90
[477] 141 116  86 112  93 355  99 101 190 145  84 111 425 146 115 122 169 104 124 115  95 128 177 164  95 525 124  89
[505] 114  85 259 170 103 167 143 375 121 120 113 119 119 326 119  91 121 271 118  92  91 113 315  86 133 143  94  86
[533] 301 109 115 165 121 189  96 363  97 166 103 124 124 150  87  92 164  94 126 198  83  86 120 136 111 123 233 109
[561] 141 483  97 125  90 122 110 263 116 113 113 242  96 107 117 142 181
```

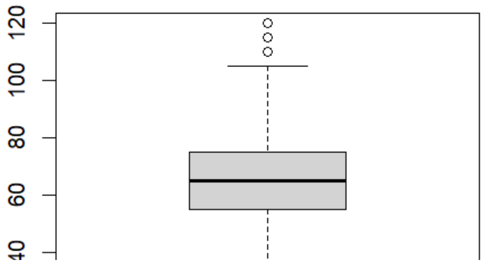
- **Box Plot**



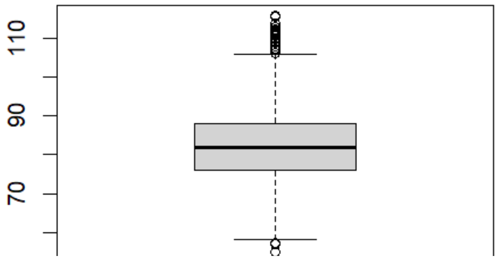
boxplot(age) 1



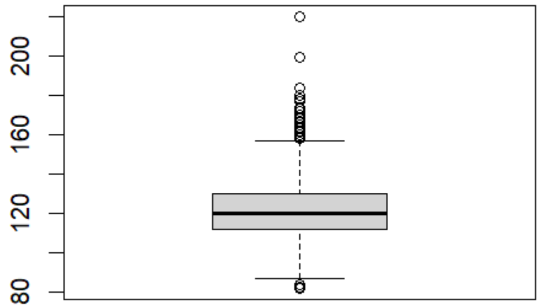
boxplot(height.cm) 1



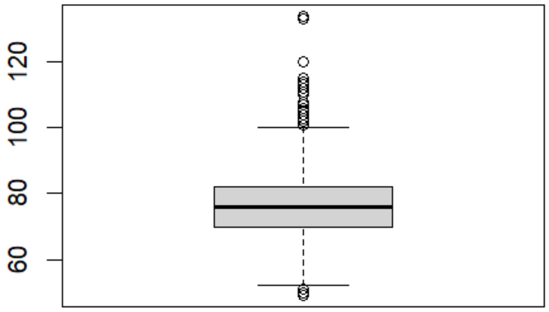
boxplot(weight.kg.) 1



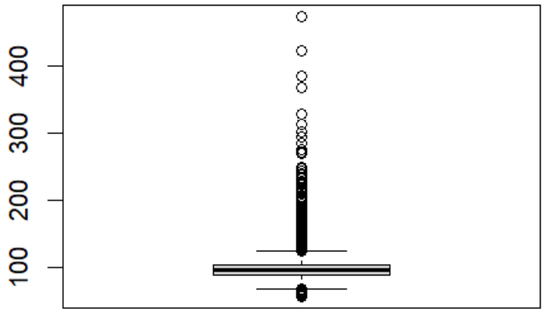
boxplot(waist.cm.) 1



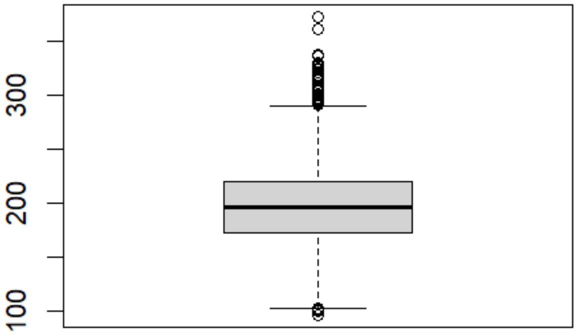
boxplot(systolic) 1



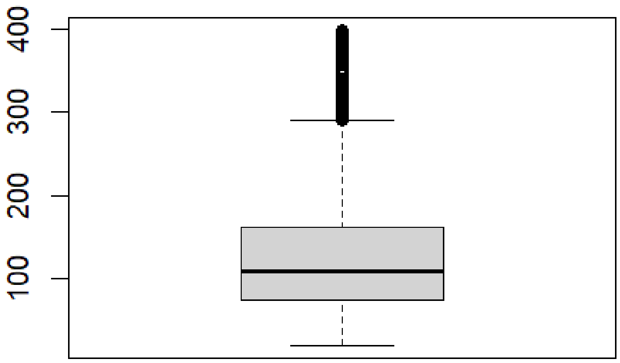
boxplot(relaxation) 1



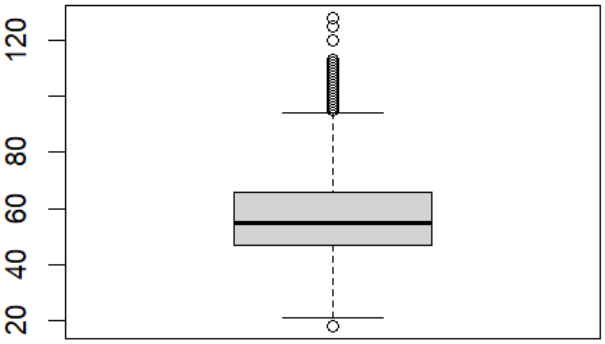
boxplot(fasting.blood.sugar) 1



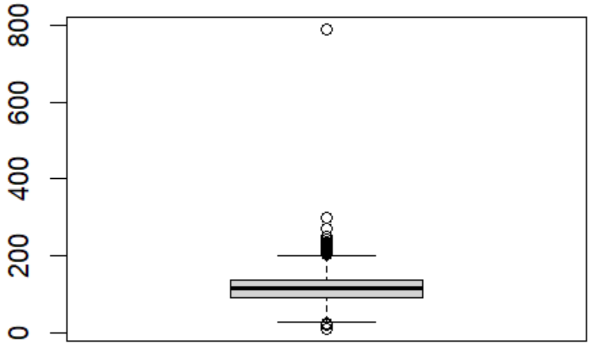
boxplot(Cholesterol) 1



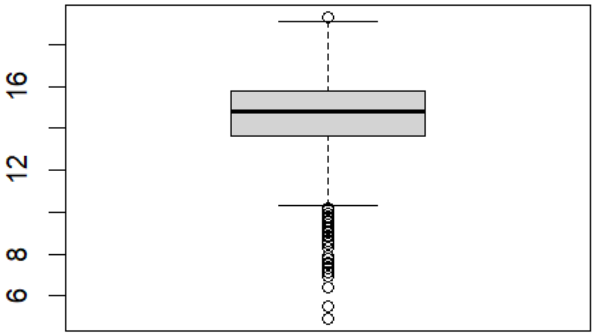
boxplot(triglyceride) 1



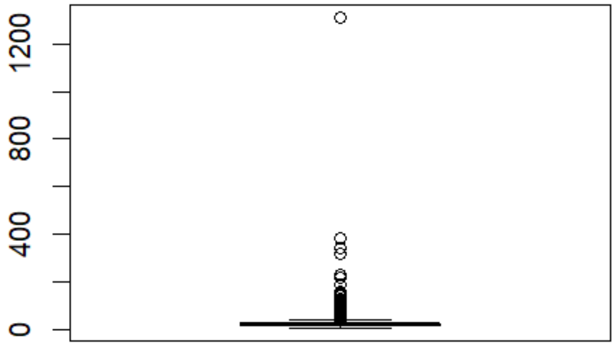
boxplot(HDL) 1



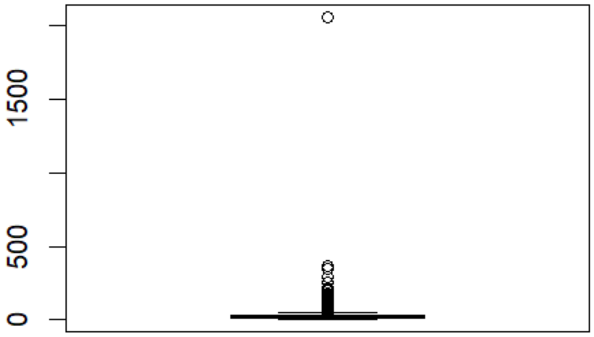
boxplot(LDL) 1



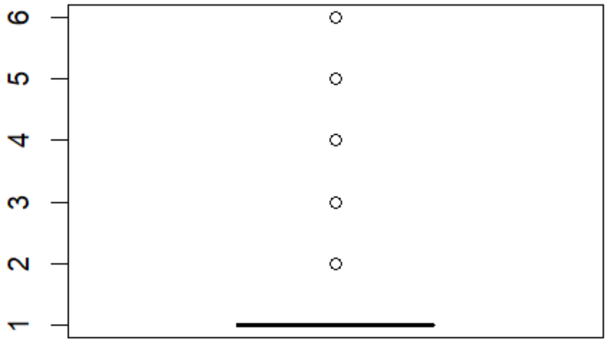
boxplot(hemoglobin) 1



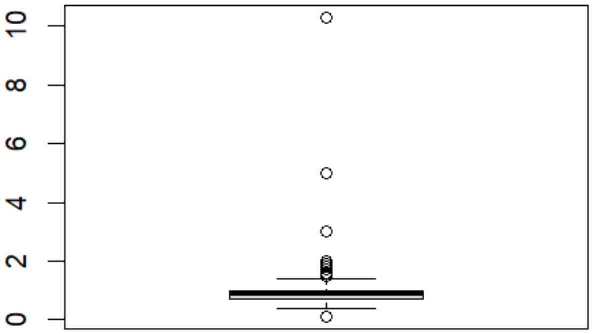
boxplot(Urine.protein) 1



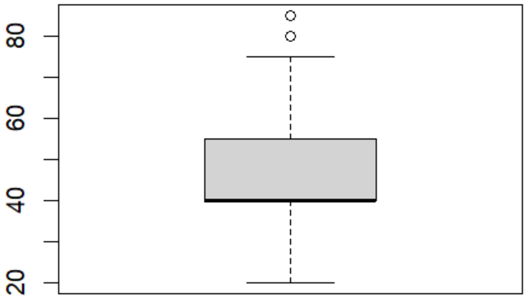
boxplot(serum.creatinine) 1



boxplot(AST) 1

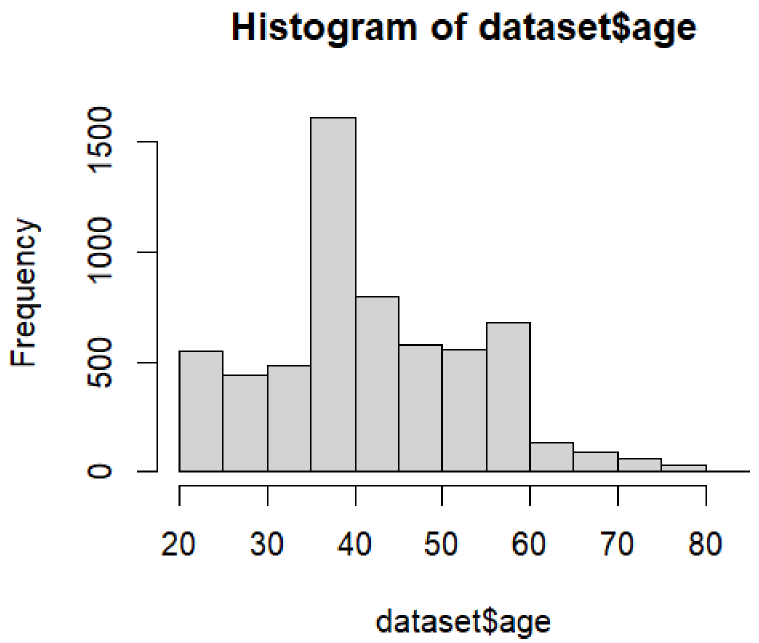


boxplot(ALT) 1

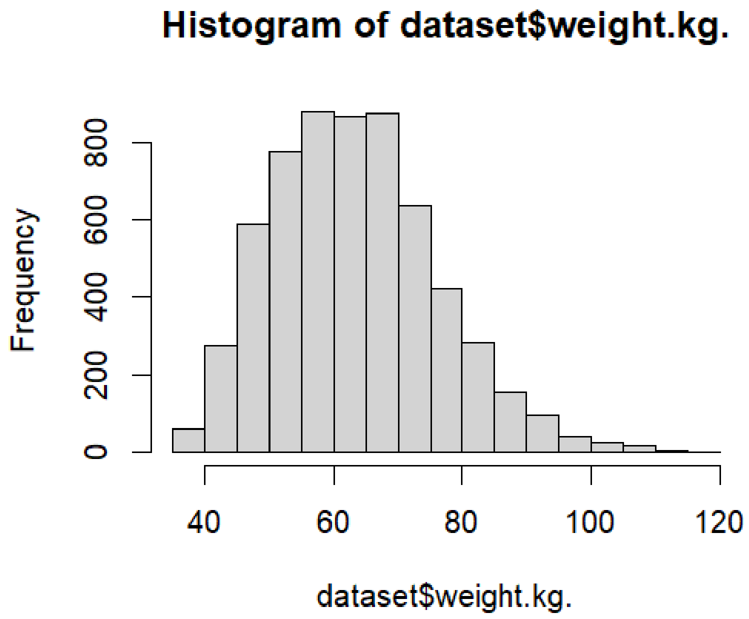


boxplot(Gtp) 1

- **Histogram**

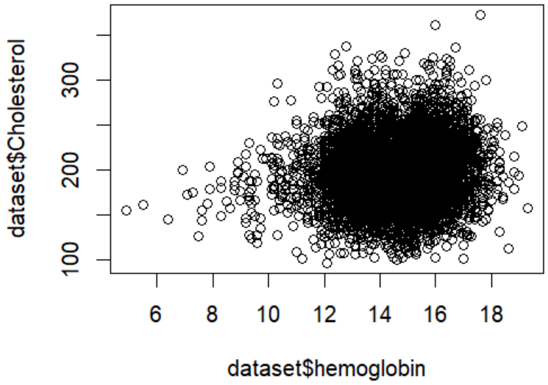


Description : The chart shows the frequency of age attribute. It shows that the most frequent ages are between (30-40) years old.



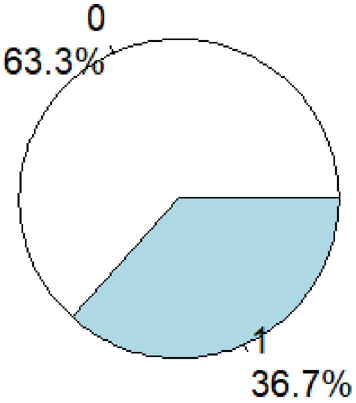
Description : The chart shows the frequency of weight attribute. It shows that the most frequent weights are between (45-75) kg.

- **Scatter**



Description : As we observed, this scatter plot shows the gathering spot of the values of the attribute "hemoglobin"and "Cholesterol" were correlated . Also, we can see some values that are far away could be detected as outliers.

- **Pie Chart**

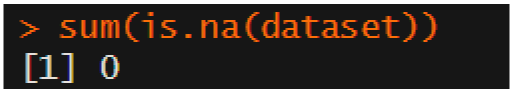


Description :The chart shows the frequency of smoke attribute, and it shows that more than 50% do not smoke.

After calculating the five number summary, box plots and outliers, we noticed that our dataset contains a lot of outliers. In this case, our dataset needs preprocessing (cleaning) to remove the outliers, checking nulls, encoding, and normalization, Correlation, Discretization.

4 Data preprocessing

- **isNull**



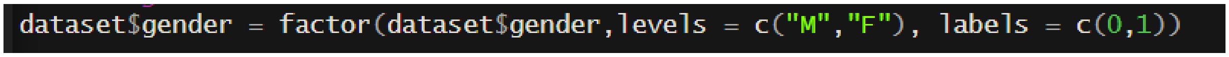
```
> sum(is.na(dataset))
[1] 0
```

no null value has been found un the dataset.

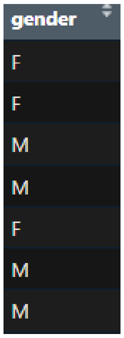
- **Encoding**

For data encoding,some attributes were already encoded, like the (smoke, dental caries) attribute, so we only encoded(gender,oral,tartar) attribute to make the data easy to understand and easy when we did the classification method.

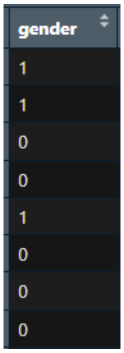
Gender



```
dataset$gender = factor(dataset$gender,levels = c("M","F"), labels = c(0,1))
```

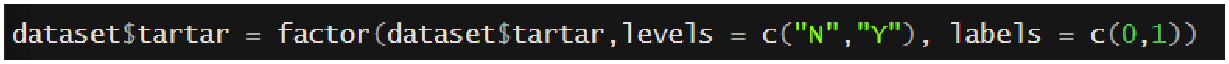


gender
F
F
M
M
F
M
M

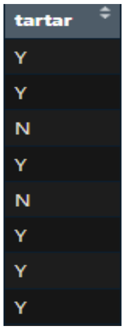


gender
1
1
0
0
1
0
0
0

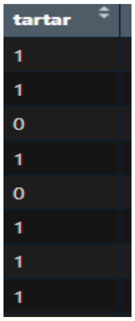
Tartar



```
dataset$tartar = factor(dataset$tartar,levels = c("N","Y"), labels = c(0,1))
```

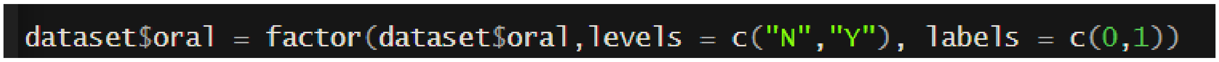


tartar
Y
Y
N
Y
N
Y
Y
Y

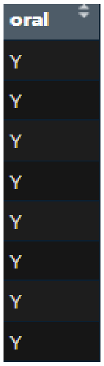


tartar
1
1
0
1
0
1
1
1

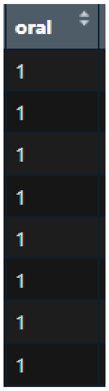
Oral



```
dataset$oral = factor(dataset$oral,levels = c("N","Y"), labels = c(0,1))
```



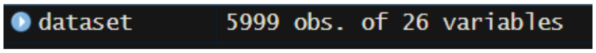
oral
Y
Y
Y
Y
Y
Y
Y
Y



oral
1
1
1
1
1
1
1
1

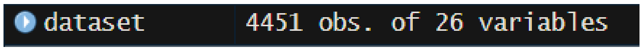
- **Replacing and removing the outliers**

before Replacing and removing the outliers



```
dataset          5999 obs. of 26 variables
```

after Replacing and removing the outliers



```
dataset          4451 obs. of 26 variables
```

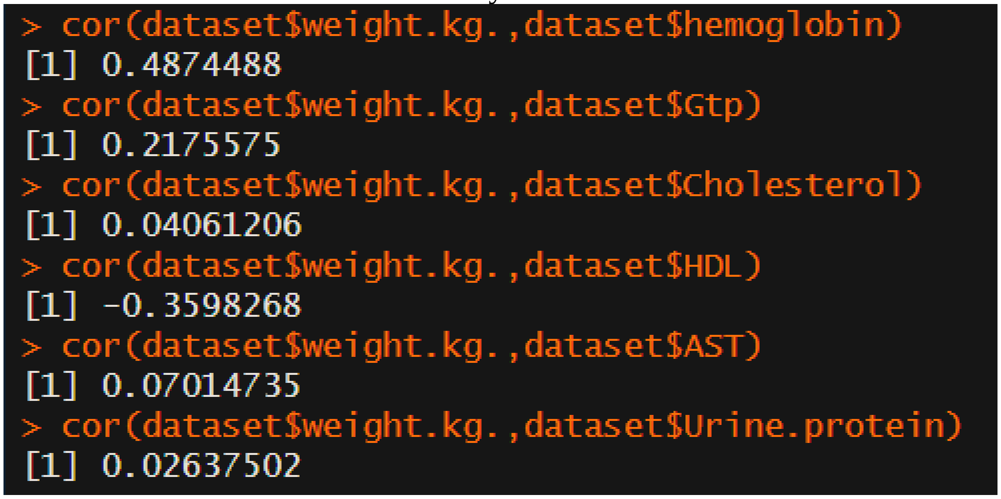
After replacing the outliers with the mean, we removed the outliers since they are increasing the variability in the data, but even after removing the outliers we don’t have 100% clean data because we have a lot of outliers .

- **Normalization**

Our data do not need normalization because there is no significant distance between the attributes so we don't need to normalize it.

- **Correlation analysis**

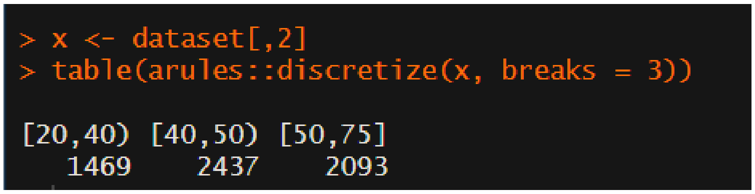
correlation analysis has been used to determine the relation of each two attributes, correlation can measure how strongly one attribute implies the other, and how they are dependent or independent of each other based on the available data. If the value is close to 1 then that means that the attributes are dependent on each other and if it is close to 0 then that means they are independent. and correlation can be positive or negative.
so we test some attributes to see if they are correlated to each other or not.



```
> cor(dataset$weight.kg.,dataset$hemoglobin)
[1] 0.4874488
> cor(dataset$weight.kg.,dataset$Gtp)
[1] 0.2175575
> cor(dataset$weight.kg.,dataset$Cholesterol)
[1] 0.04061206
> cor(dataset$weight.kg.,dataset$HDL)
[1] -0.3598268
> cor(dataset$weight.kg.,dataset$AST)
[1] 0.07014735
> cor(dataset$weight.kg.,dataset$Urine.protein)
[1] 0.02637502
```

So we figure that the weight attribute is positively dependent on the hemoglobin and GTP and negatively dependent on the HDL, and the rest of the attributes are in between.

- **Discretization**



```
> x <- dataset[,2]
> table(arules::discretize(x, breaks = 3))

[20,40) [40,50) [50,75]
   1469    2437    2093
```

We discretization the age attribute, and we found that from age 20 to less than 40 there are 1469 records , from age 40 to less than 50 there are 2437 records and from age 50 to 75 there are 2093 records,after that we found that most of out dataset is people from the ages 40-50 .

5 Data Mining Technique

Classification:
We used the decision tree method for the classification ,because we have a class label attribute which is smoking by predicting if the person smokes or not through their bio-signals. We will divide our dataset by applying a binary tree into “training dataset” and “test dataset”, we applied these packages: (party, e1071, caret) ,using the following methods: ctree , predict and confusionMatrix. Package: party, e1071, caret.

Clustering:
We used the K-means technique, because most of our data is numeric, and we transformed the non-numeric data to numeric, to do the K-mean method,that represents the clusters by the center of the cluster. K-means select randomly k objects as clusters and assign the objects to the nearest cluster center. We used these packages: cluster , factoextra , NbClust , magrittr , GGally, plotly, using the following methods: Scale, Kmeans, Fviz_cluster, silhouette, Fviz_nbclust, ggparcoord().

6 Training procedure

- Classification:

1. Training set of 70% and testing set of 30%
    - First: we need to set the seed we pick seed(1234)

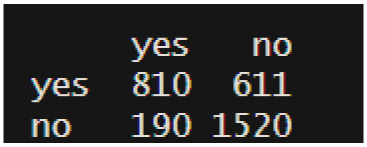
```
set.seed(1234)
```

- Second: we need the split the data to training and teasting set

```
ind <- sample(2,nrow(data),replace=TRUE,prob = c(0.7,0.3))
trainData <-data[ind==1,]
testData <-data[ind==2,]
myFormula <- smoking ~ gender      +age+  height.cm.      +weight.kg.+   waist.cm.
        +eyesight.left.+        eyesight.right.+        hearing.left.+  hearing.right.+
        systolic        +relaxation+  fasting.blood.sugar+  Cholesterol+  triglyceride+
        HDL+ LDL+ hemoglobin+  Urine.protein+serum.creatinine+      AST+  ALT
        +Gtp+ dental.caries+ tartar
```

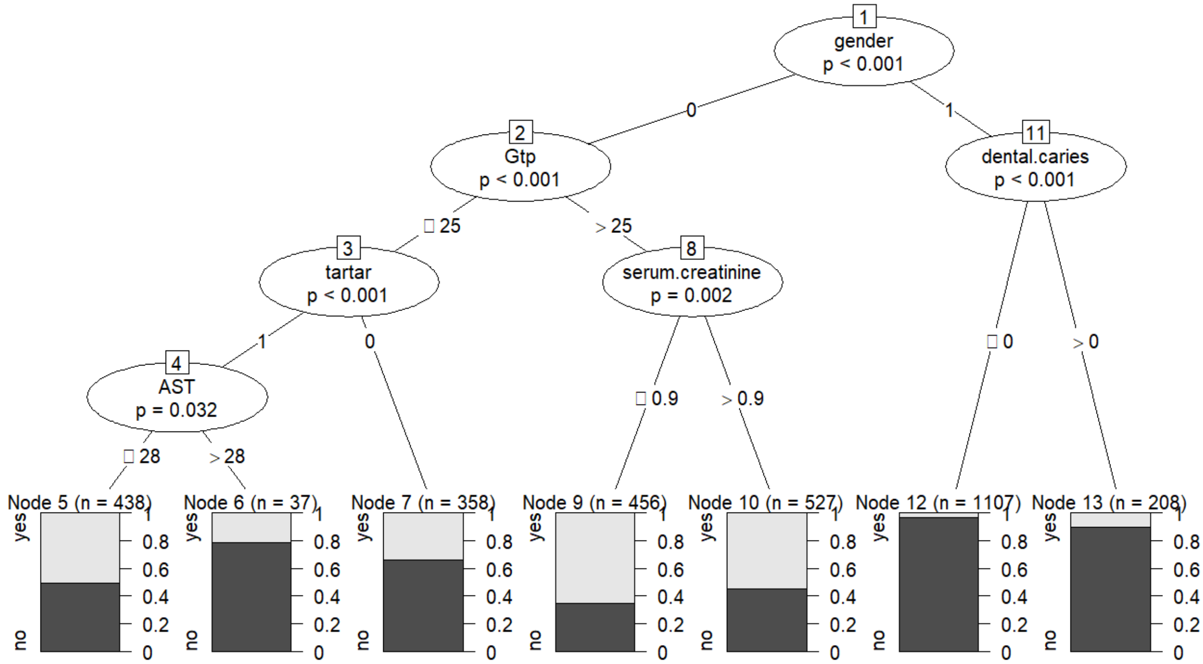
- Thired: we print the print the predict table for the train data

```
table(predict(dataset_ctree), trainData$smoking)
```

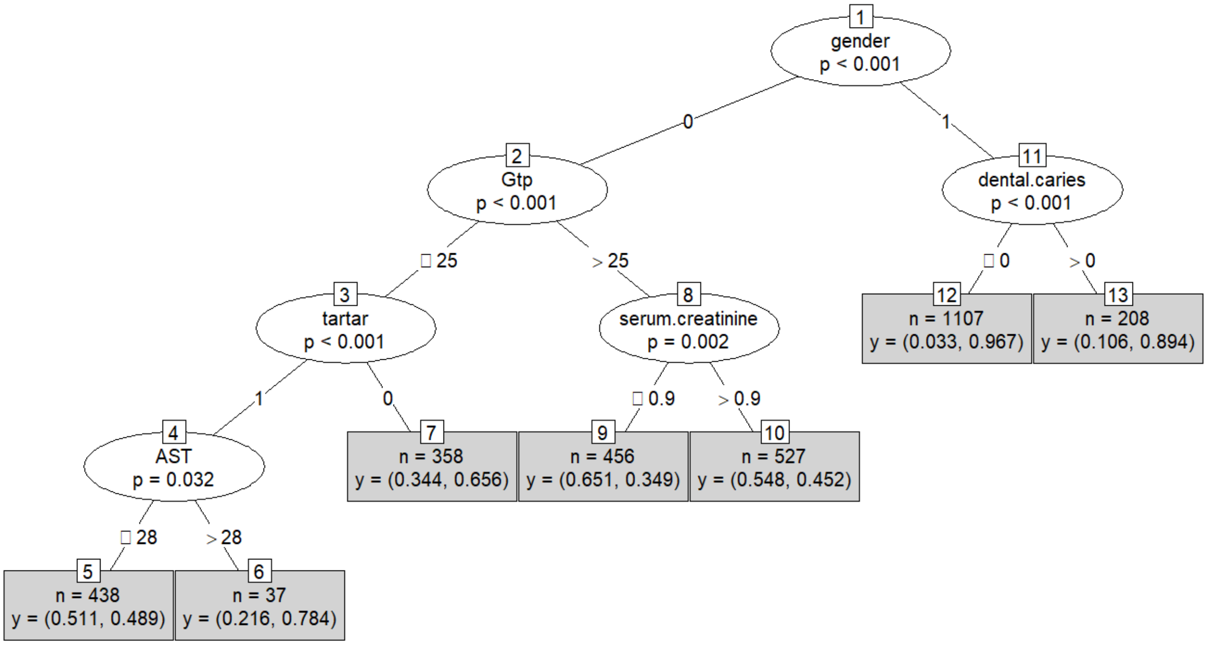


```
      yes   no
yes   810  611
no    190 1520
```

- Fourth: we print the Decision tree



Decision tree (70%,30%)



SimpleDecision tree (70%,30%)

2. Training set of(80%,20%)

   - First: we need to set the seed we pick seed(1234)

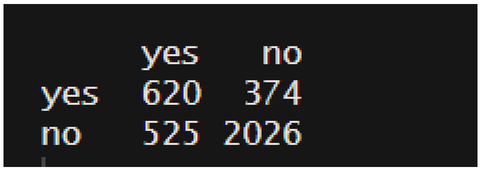
set.seed(1234)

   - Second: we need the split the data to training and testing set

ind <- sample(2,nrow(data),replace=TRUE,prob = c(0.8,0.2))
trainData <-data[ind==1,]
testData <-data[ind==2,]
myFormula <- smoking ~ gender +age+ height.cm. +weight.kg.+ waist.cm.
+eyesight.left.+ eyesight.right.+ hearing.left.+ hearing.right.+
systolic +relaxation+ fasting.blood.sugar+ Cholesterol+ triglyceride+
HDL+ LDL+ hemoglobin+ Urine.protein+serum.creatinine+ AST+ ALT
+Gtp+ dental.caries+ tartar

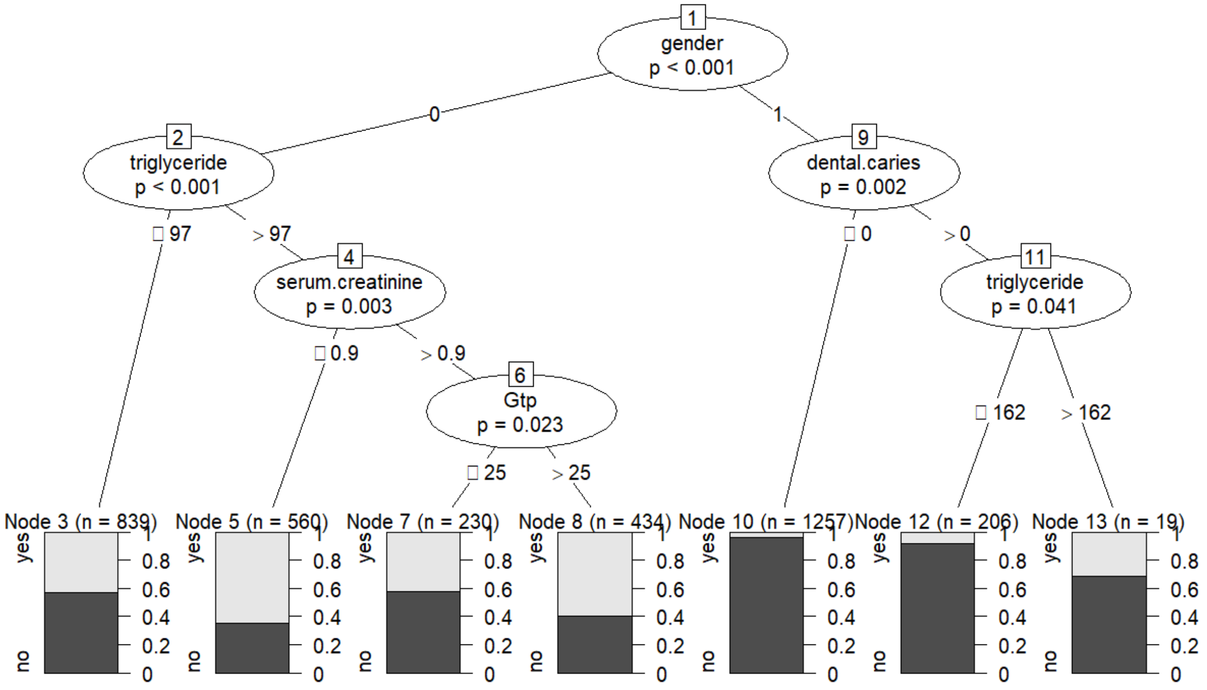
   - Thired: we print the print the predict table for the train data

table(predict(dataset_ctree), trainData$smoking)

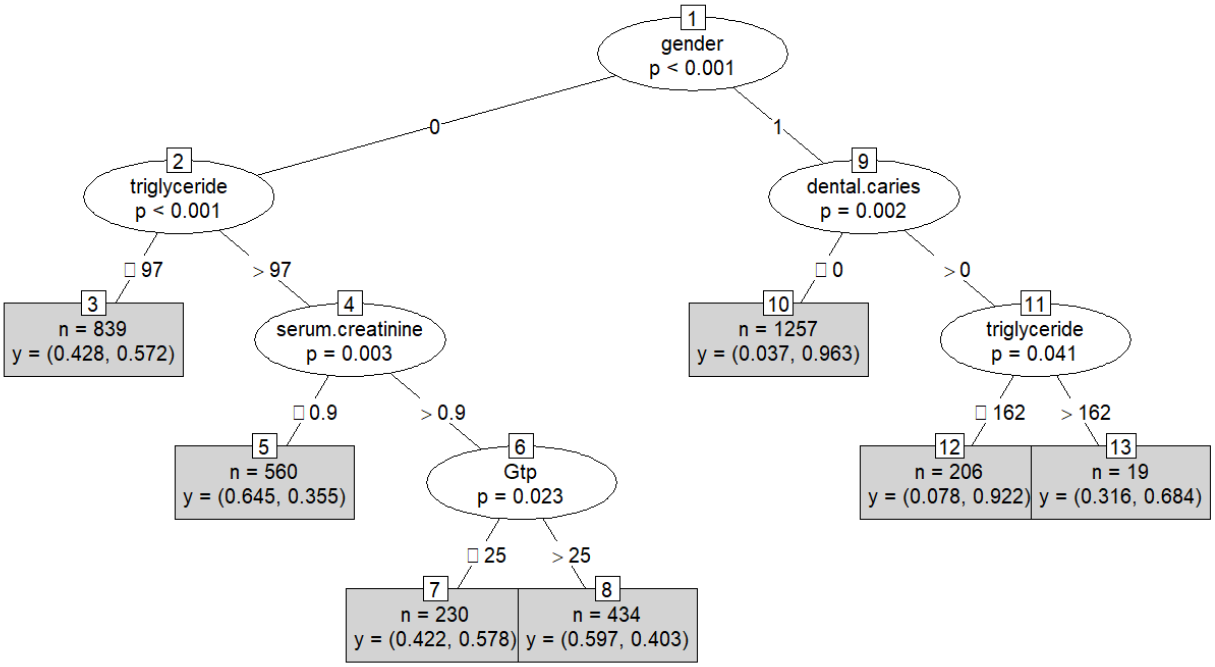


```
      yes   no
yes   620  374
no    525 2026
```

   - Fourth: we print the Decision tree



Decision tree (80%,20%)



SimpleDecision tree (80%,20%)

3. Training set of(60%,40%)

   - First: we need to set the seed we pick seed(1234)

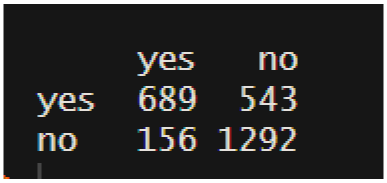
```
set.seed(1234)
```

- Second: we need the split the data to training and testing set

```
ind <- sample(2,nrow(data),replace=TRUE,prob = c(0.6,0.4))
trainData <-data[ind==1,]
testData <-data[ind==2,]
myFormula <- smoking ~ gender +age+ height.cm. +weight.kg.+ waist.cm. +eyesight.left.+ eyesight.right.+ hearing.left.+ hearing.right.+ systolic +relaxation+ fasting.blood.sugar+ Cholesterol+ triglyceride+ HDL+ LDL+ hemoglobin+ Urine.protein+serum.creatinine+ AST+ ALT +Gtp+ dental.caries+ tartar
```

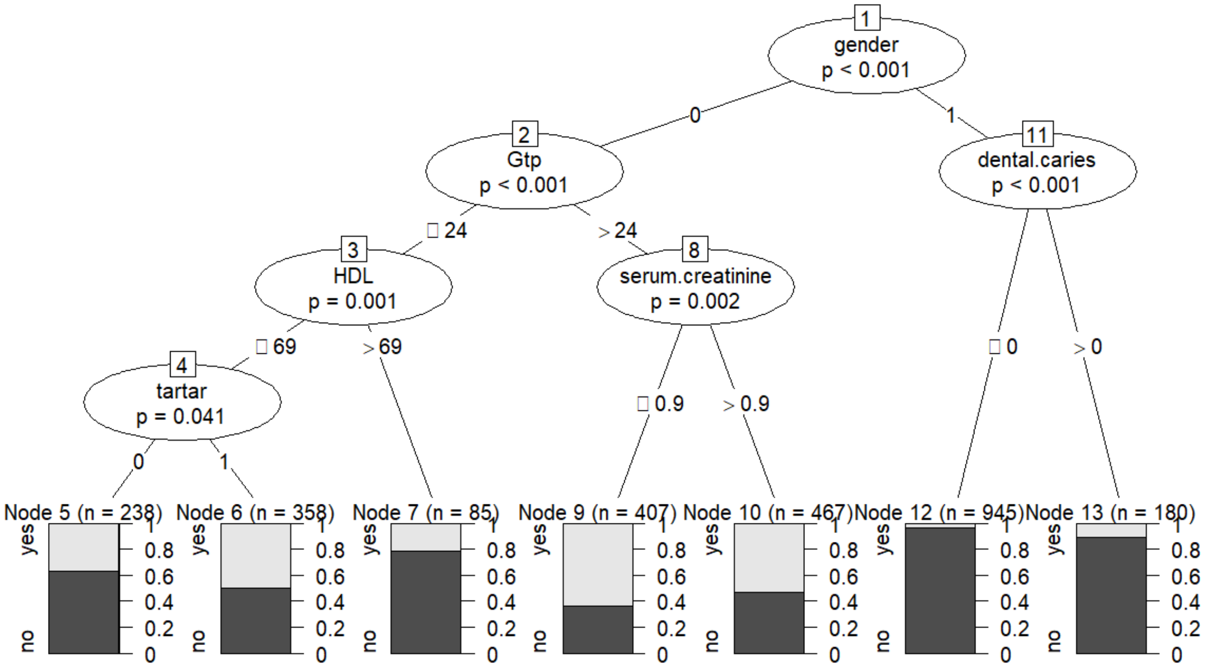
- Thired: we print the print the predict table for the train data

```
table(predict(dataset_ctree), trainData$smoking)
```

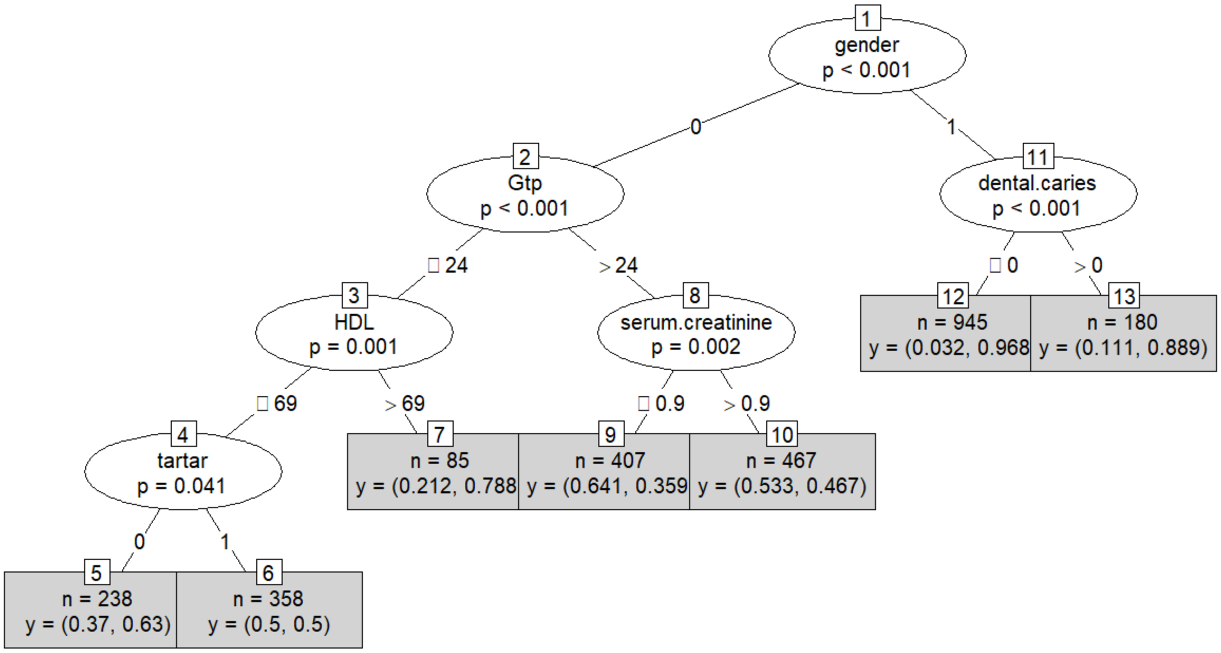


```
      yes   no
yes   689  543
no    156 1292
```

- Fourth: we print the Decision tree



Decision tree (60%,40%)



SimpleDecision tree (60%,40%)

- Clustering:
- First: we transform the non-numeric to numeric data

```
dataset$gender<-as.numeric(dataset$gender)
dataset$tartar<-as.numeric(dataset$tartar)
dataset$oral<-as.numeric(dataset$oral)
```

- Second: after scaling there is an attribute with null values in all the row so we have to remove it

```
dataset2 <- dataset[,-23]
```

- Thired: data types should be transformed into numeric types before clustering.

```
dataset2 <- scale(dataset2)
```

- Fourth: k-means clustering set a seed for random number generation to make the results

```
set.seed(9000)
```

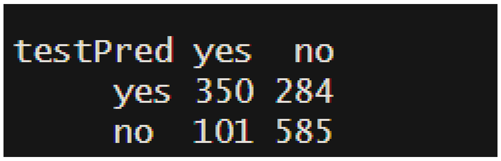
7 Evaluation and Comparison

- Classification:

for the classification we use the confusion matrix method to evaluate the test data.

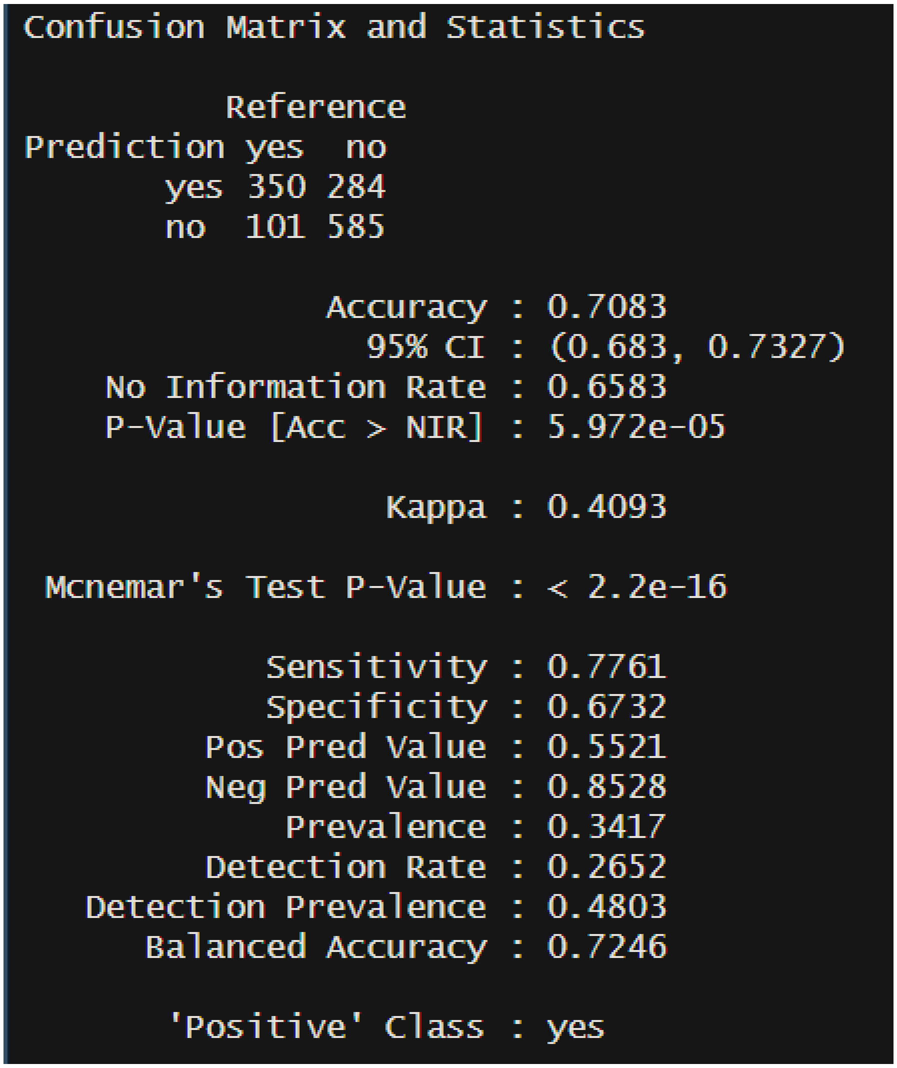
	(70%,30%)	(80%,20%)	(60%,40%)
Accuracy	70.83%	69.09%	72.33%
precision	55.20%	54.28%	56.90%
sensitivity	77.61%	51.63%	78.88%
specificity	67.32%	78%	68.93%

1. Training set of 70% and testing set of 30%
   - predict table for the test data



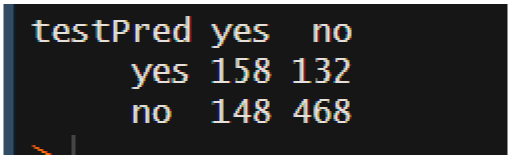
```
testPred yes  no
     yes 350 284
     no  101 585
```

- Confusion Matrix and Statistics



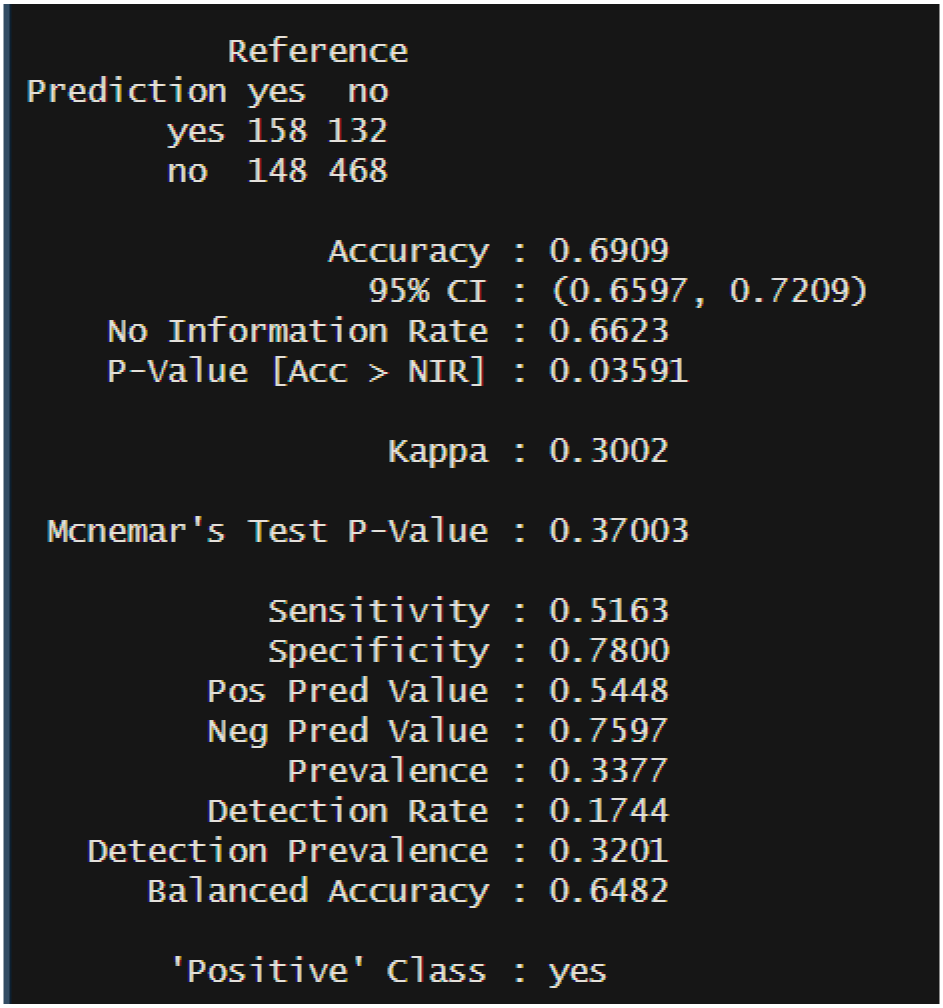
```
Confusion Matrix and Statistics

          Reference
Prediction yes  no
       yes 350 284
       no  101 585

               Accuracy : 0.7083
                 95% CI : (0.683, 0.7327)
    No Information Rate : 0.6583
    P-Value [Acc > NIR] : 5.972e-05

                  Kappa : 0.4093

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7761
            Specificity : 0.6732
         Pos Pred Value : 0.5521
         Neg Pred Value : 0.8528
             Prevalence : 0.3417
         Detection Rate : 0.2652
   Detection Prevalence : 0.4803
      Balanced Accuracy : 0.7246

       'Positive' Class : yes
```

2. Training set of 80% and testing set of 20%
    - predict table for the test data



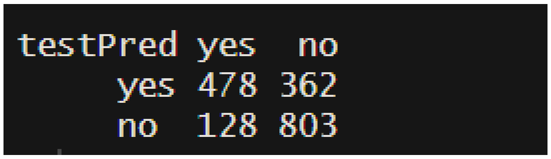
```
testPred yes  no
     yes 158 132
     no  148 468
```

    - Confusion Matrix and Statistics



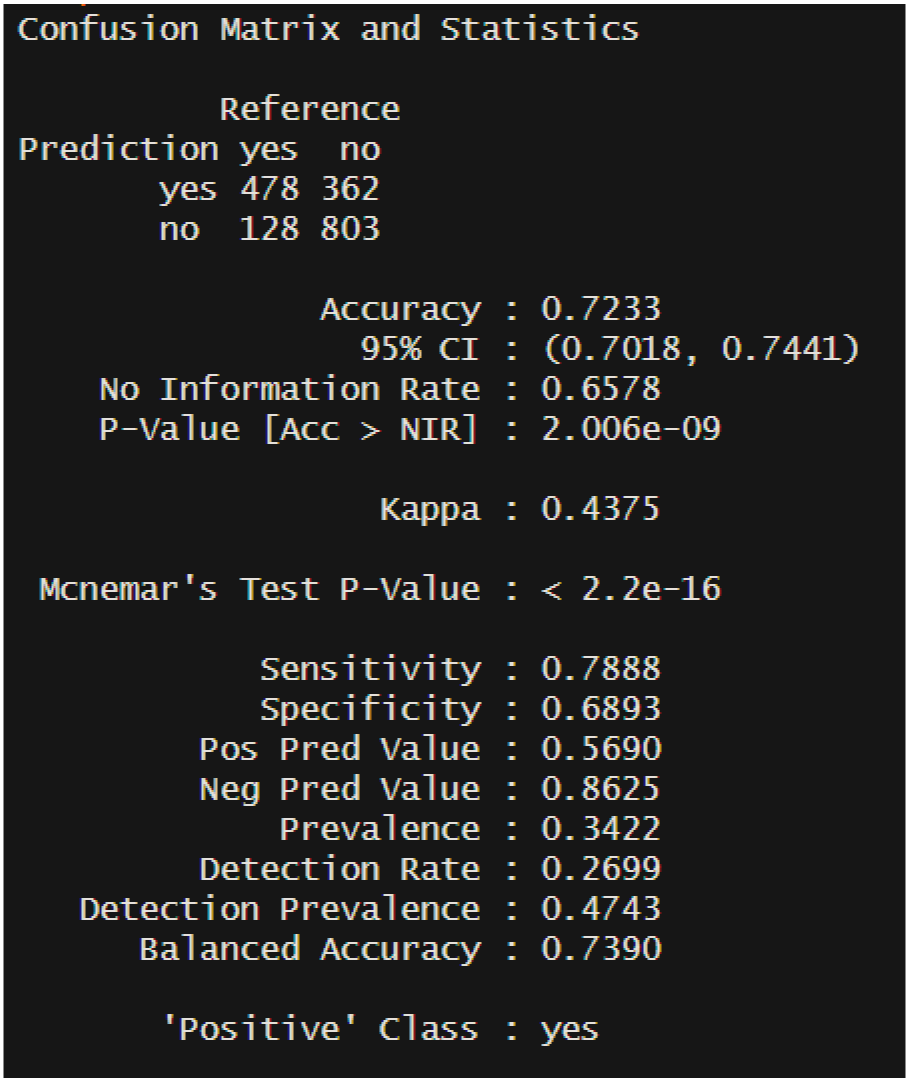
```
          Reference
Prediction yes  no
       yes 158 132
       no  148 468

               Accuracy : 0.6909
                 95% CI : (0.6597, 0.7209)
    No Information Rate : 0.6623
    P-Value [Acc > NIR] : 0.03591

                  Kappa : 0.3002

 Mcnemar's Test P-Value : 0.37003

            Sensitivity : 0.5163
            Specificity : 0.7800
         Pos Pred Value : 0.5448
         Neg Pred Value : 0.7597
             Prevalence : 0.3377
         Detection Rate : 0.1744
   Detection Prevalence : 0.3201
      Balanced Accuracy : 0.6482

       'Positive' Class : yes
```

3. Training set of 60% and testing set of 40%
    - predict table for the test data



```
testPred yes  no
     yes 478 362
     no  128 803
```

    - Confusion Matrix and Statistics

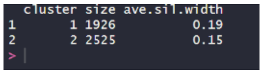
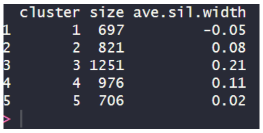
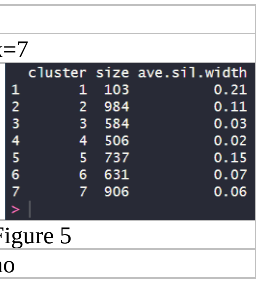


```
Confusion Matrix and Statistics

          Reference
Prediction yes  no
       yes 478 362
       no  128 803
                                          
               Accuracy : 0.7233          
                 95% CI : (0.7018, 0.7441)
    No Information Rate : 0.6578          
    P-Value [Acc > NIR] : 2.006e-09       
                                          
                  Kappa : 0.4375          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7888          
            Specificity : 0.6893          
         Pos Pred Value : 0.5690          
         Neg Pred Value : 0.8625          
             Prevalence : 0.3422          
         Detection Rate : 0.2699          
   Detection Prevalence : 0.4743          
      Balanced Accuracy : 0.7390          
                                          
       'Positive' Class : yes             
```

- Clustering:

for the clustering we use silhouette width to evaluate the best clustering number of k cluster.

- attributes that we use : gender ,age, height.cm., weight.kg. , waist.cm. , eyesight.left. , eyesight.right., hearing.left.,hearing.right. ,systolic, relaxation ,fasting.blood.sugar, Cholesterol ,triglyceride ,HDL ,LDL, hemoglobin, serum.creatinine ,AST ,ALT,Gtp, dental.caries, tartar, smoking.
- attributes that we did not use : oral , Urin.protein.
- reason : because when we scale the dataset they become null values so we can not use k-mean if there is any null value.

K-mean			
num of K	k=2	k=5	k=7
Average silhouette width	 cluster size ave.sil.width 1 1 1926 0.19 2 2 2525 0.15 >	 cluster size ave.sil.width 1 1 697 -0.05 2 2 821 0.08 3 3 1251 0.21 4 4 976 0.11 5 5 706 0.02 >	 cluster size ave.sil.width 1 1 103 0.21 2 2 984 0.11 3 3 584 0.03 4 4 506 0.02 5 5 737 0.15 6 6 631 0.07 7 7 906 0.06 >
Visualization	Figure 1	Figure 3	Figure 5
Is Optimal ?	yes	no	no

1. k=2

cluster plot:

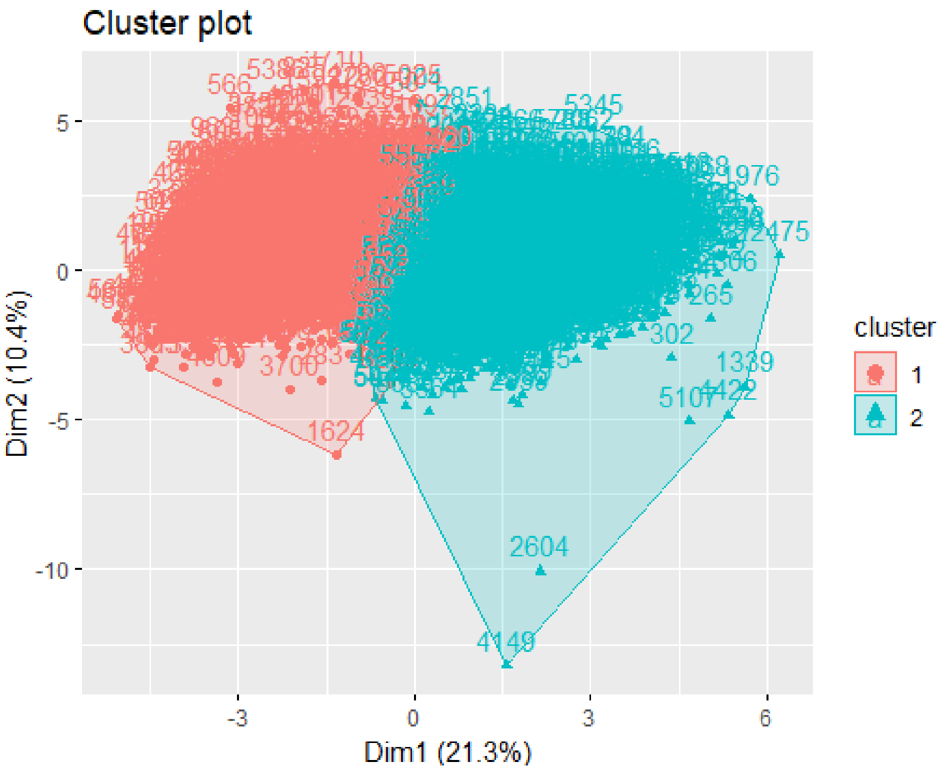


Figure 1

cluster silhouette plot :

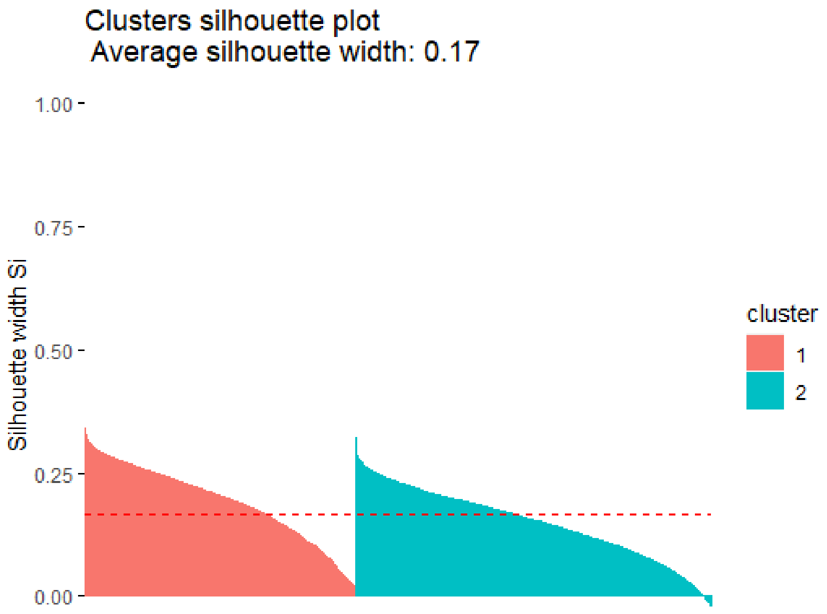


Figure 2

2. k=5

cluster plot :

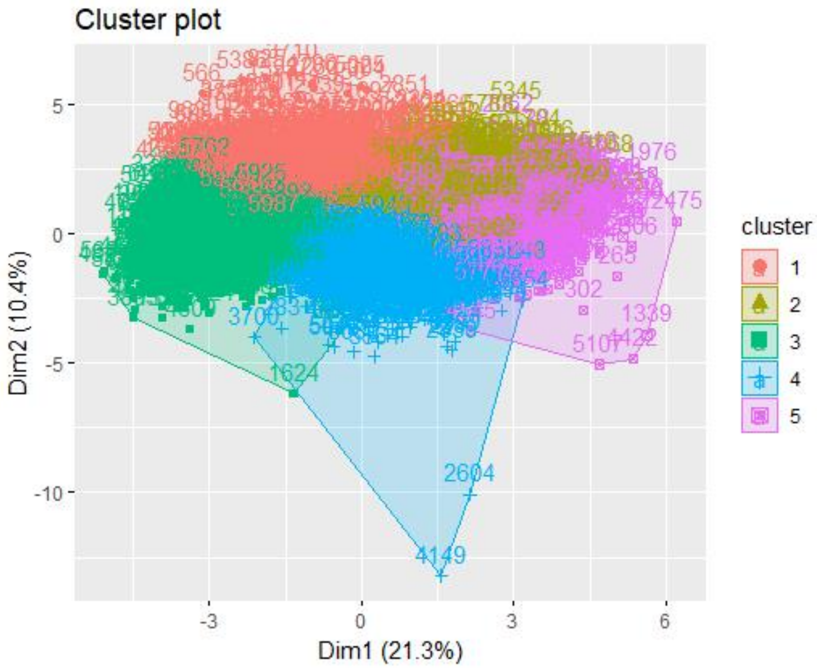


Figure 3

cluster silhouette plot :

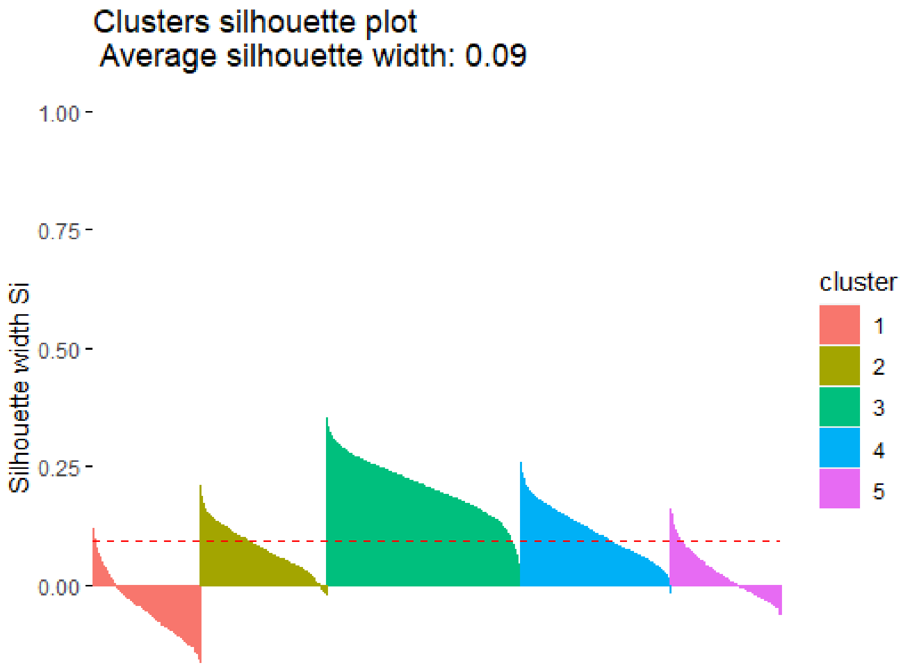


Figure 4

3. k=7

cluster plot :

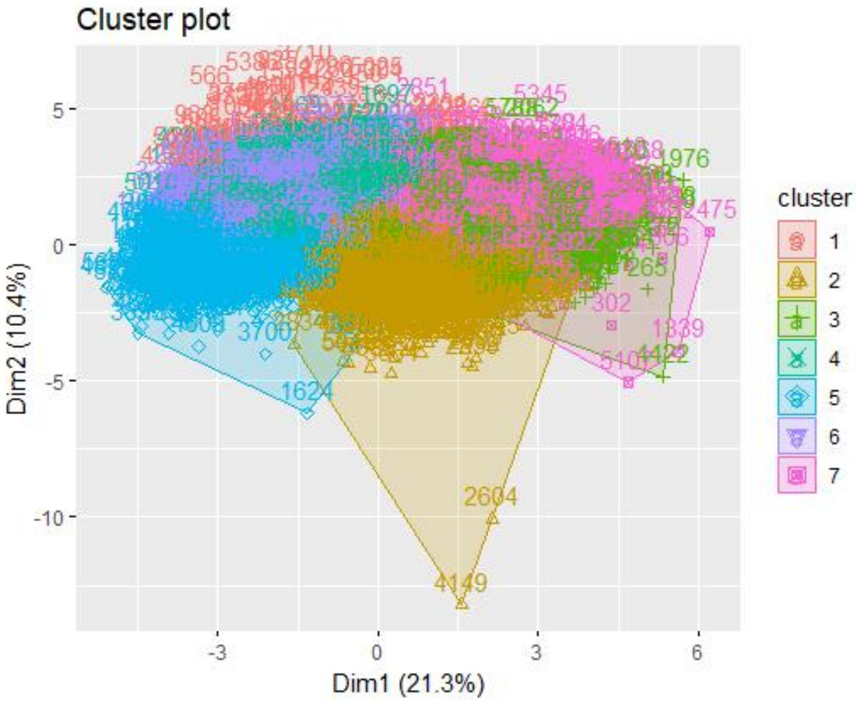


Figure 5

cluster silhouette plot :

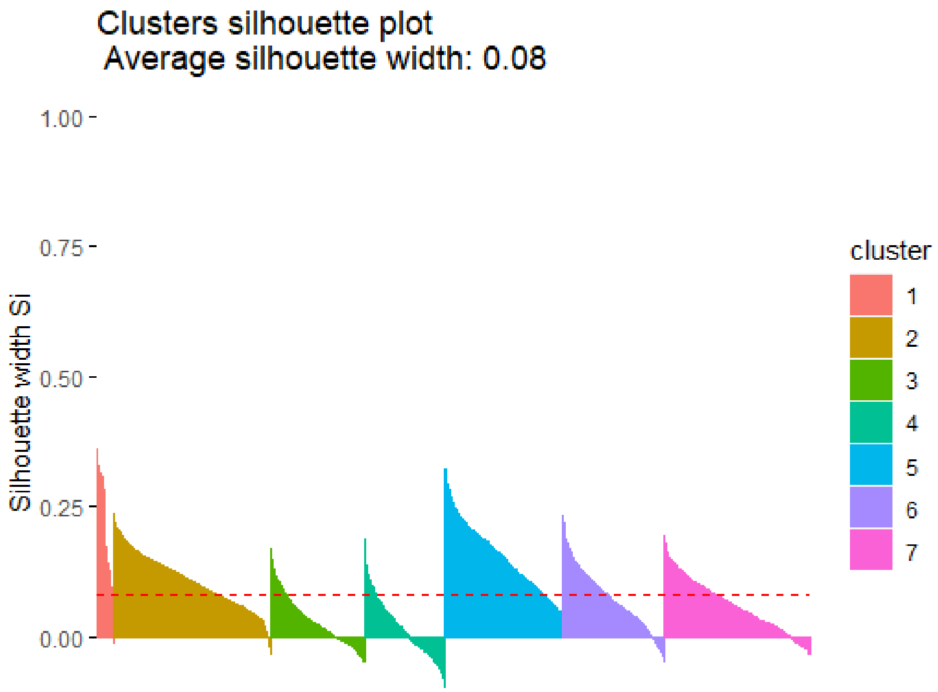


Figure 6

4. Silhouette width for all clusters:

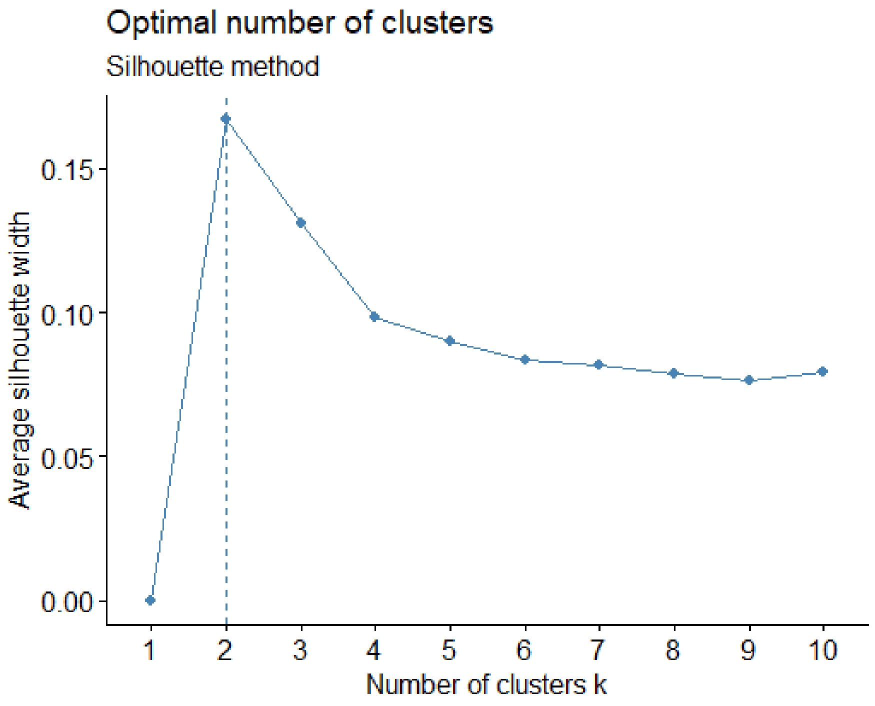


Figure 7

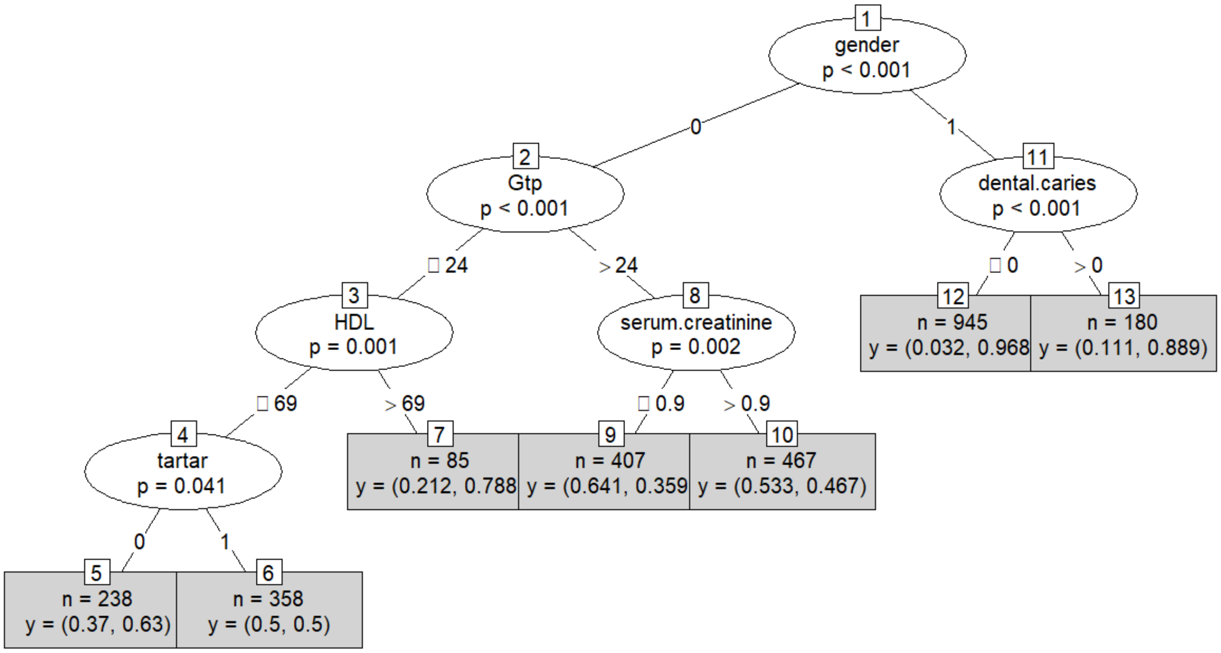
8 Findings:

Here we are talking about the results after implementing the classification and clustering.

Classification:
concisely, we studied how attributes affect each other by mining the data by using classification technique and tree model. Use ctree, predict, and confusionMatrix.
So, we applied the tree algorithm on 3 different sizes, the first one is training set 70% and testing set 30%. The second, training set 80% and testing set 20%, and the last is 60% training 40% testing.

After cleaning our data and removing the outliers we still did not get a 100% clean data, on according of that, after seeing the results, we concluded that the best result of accuracy and other evaluation metrics could be in our test is the last one which we divided into (60%,40%), which was constructed by using 72.33% of dataset tuples as training data as a result it learned better than other models.

According to the decision tree, it is divided based on the gender variable since it has the highest information gain and that helped us know the last decision which is male smokers are more than female. Well, this kind of prediction will become useful for hospitals to help them with knowing their patients better. Moreover, companies or any interested individual may use it for any benefits.



Clustering:

We implemented the plotting above and picked out three random numbers which are 2,5,7. After using fviz_cluster() to get the whole cluster we used the silhouette() method to give us the average of each cluster in the whole cluster. We also accepted the optimal number 2 because it is the closest number to 1.

While the K=2 the average of the cluster would be equal to 0.17 which was the best average result we got in our test, in each cluster silhouette are 0.19 for the first cluster and 0.15 for the second. From the centers of clusters we can describe the relation between attributes and each cluster. Within K=2 the state in cluster 1 has 1926 observations , in cluster 2 we have 2525 observations and this cluster has the minimum sum of squares that equals 16.7%,cluster2 it is for males and we notice that they smoke more than cluster1 which is cluster for females.

When the K=5 the average of cluster 0.09, and in each cluster silhouette are 0.05 for first cluster and 0.08 for the second cluster 0.21 for the third cluster 0.11 for the fourth cluster and 0.02 for the fifth cluster. because the total sum of squares is 26.6% and it gives unclear plotting, so we can not analyze it.

When the K=7 the average of cluster 0.08, and in each cluster silhouette are 0.21 for first cluster and 0.11 for second cluster 0.03 for the third cluster 0.02 for the fourth cluster 0.15 for the fifth cluster and 0.07 the sixth cluster 0.06 for the seventh cluster . And the total sum of squares is 32.8 %so, it's unclear so, the result is not analyzable.

In conclusion, when the K=2 the average width is 0.17 it is good because K-means consider the number close to 1 is better than other.And we verified which is the best number using fviz_nbclus() method that gives number 2. Also, saw the silhouette averages are greater than 0 that means each observation is well clustered.

9 Code

we use all the attrubets in the dataset.

- preprocessing :

```
#Encoding
dataset$gender = factor(dataset$gender,levels = c("M","F"), labels = c(0,1))

dataset$tartar = factor(dataset$tartar,levels = c("N","Y"), labels = c(0,1))

dataset$oral = factor(dataset$oral,levels = c("N","Y"), labels = c(0,1))
 View(dataset)
#outliers
#After replacing and removing outliers:


#age

boxplot.stats(dataset$age)$out   ###before
minVal <- boxplot.stats(dataset$age)$stats[1]
maxVal <- boxplot.stats(dataset$age)$stats [5]
myValue <- mean(dataset$age)
```

```
dataset [dataset$age < minVal | dataset$age > maxVal, "age"] <- myValue
boxplot(dataset$age)
boxplot.stats(dataset$age)$out ####after
x1= boxplot.stats(dataset$age)$out ####after
boxplot(dataset$age)

####################

#blood sugar

boxplot.stats(dataset$fasting.blood.sugar)$out   ##before
minVal <- boxplot.stats(dataset$fasting.blood.sugar)$stats[1]
maxVal <- boxplot.stats(dataset$fasting.blood.sugar)$stats [5]
myValue <- mean(dataset$fasting.blood.sugar)
dataset [dataset$fasting.blood.sugar < minVal | dataset$fasting.blood.sugar > maxVal,
"fasting.blood.sugar"] <- myValue
boxplot(dataset$fasting.blood.sugar)
boxplot.stats(dataset$fasting.blood.sugar)$out

outliers <- boxplot(dataset$fasting.blood.sugar, plot=FALSE)$out
dataset <- dataset[-which(dataset$fasting.blood.sugar %in% outliers),]
boxplot(dataset$fasting.blood.sugar)
boxplot.stats(dataset$fasting.blood.sugar)$out ###after
###########################################################

#height.cm.

boxplot.stats(dataset$height.cm.)$out   ##before
minVal <- boxplot.stats(dataset$height.cm.)$stats[1]
maxVal <- boxplot.stats(dataset$height.cm.)$stats [5]
myValue <- mean(dataset$height.cm.)
dataset [dataset$height.cm. < minVal | dataset$height.cm. > maxVal, "height.cm."] <- myValue
boxplot(dataset$height.cm.)
boxplot.stats(dataset$height.cm.)$out #after
########################################################

#weight.kg.

boxplot.stats(dataset$weight.kg.)$out   ##before
minVal <- boxplot.stats(dataset$weight.kg.)$stats[1]
maxVal <- boxplot.stats(dataset$weight.kg.)$stats [5]
myValue <- mean(dataset$weight.kg.)
dataset [dataset$weight.kg. < minVal | dataset$weight.kg. > maxVal, "weight.kg."] <- myValue
boxplot(dataset$weight.kg.)
boxplot.stats(dataset$weight.kg.)$out  #after
############################################################
```

```
#waist.cm.

boxplot.stats(dataset$waist.cm.)$out   ##before
minVal <- boxplot.stats(dataset$waist.cm.)$stats[1]
maxVal <- boxplot.stats(dataset$waist.cm.)$stats [5]
myValue <- mean(dataset$waist.cm.)
dataset [dataset$waist.cm. < minVal | dataset$waist.cm. > maxVal, "waist.cm."] <- myValue
boxplot(dataset$waist.cm.)
boxplot.stats(dataset$waist.cm.)$out  #after
######################################################

#systolic

boxplot.stats(dataset$systolic)$out   ##before
minVal <- boxplot.stats(dataset$systolic)$stats[1]
maxVal <- boxplot.stats(dataset$systolic)$stats [5]
myValue <- mean(dataset$systolic)
dataset [dataset$systolic < minVal | dataset$systolic > maxVal, "systolic"] <- myValue
boxplot(dataset$systolic)
boxplot.stats(dataset$systolic)$out  #after
####################################################

#relaxation

boxplot.stats(dataset$relaxation)$out   ##before
minVal <- boxplot.stats(dataset$relaxation)$stats[1]
maxVal <- boxplot.stats(dataset$relaxation)$stats [5]
myValue <- mean(dataset$relaxation)
dataset [dataset$relaxation < minVal | dataset$relaxation > maxVal, "relaxation"] <- myValue
boxplot(dataset$relaxation)
boxplot.stats(dataset$relaxation)$out

outliers <- boxplot(dataset$relaxation, plot=FALSE)$out
dataset <- dataset[-which(dataset$relaxation %in% outliers),]
boxplot(dataset$relaxation)
boxplot.stats(dataset$relaxation)$out ###after
###############################################################

#Cholesterol

boxplot.stats(dataset$Cholesterol)$out   ##before
minVal <- boxplot.stats(dataset$Cholesterol)$stats[1]
maxVal <- boxplot.stats(dataset$Cholesterol)$stats [5]
myValue <- mean(dataset$Cholesterol)
```

```
dataset [dataset$Cholesterol < minVal | dataset$Cholesterol > maxVal, "Cholesterol"] <-
myValue
boxplot(dataset$Cholesterol)
boxplot.stats(dataset$Cholesterol)$out

outliers <- boxplot(dataset$Cholesterol, plot=FALSE)$out
dataset <- dataset[-which(dataset$Cholesterol %in% outliers),]
boxplot(dataset$Cholesterol)
boxplot.stats(dataset$Cholesterol)$out ###after
############################################################

#triglyceride

boxplot.stats(dataset$triglyceride)$out   ##before
minVal <- boxplot.stats(dataset$triglyceride)$stats[1]
maxVal <- boxplot.stats(dataset$triglyceride)$stats [5]
myValue <- mean(dataset$triglyceride)
dataset [dataset$triglyceride < minVal | dataset$triglyceride > maxVal, "triglyceride"] <-
myValue
boxplot(dataset$triglyceride)
boxplot.stats(dataset$triglyceride)$out

outliers <- boxplot(dataset$triglyceride, plot=FALSE)$out
dataset <- dataset[-which(dataset$Cholesterol %in% outliers),]
boxplot(dataset$Cholesterol)
boxplot.stats(dataset$Cholesterol)$out ###after
########################################################

#HDL

boxplot.stats(dataset$HDL)$out   ##before
minVal <- boxplot.stats(dataset$HDL)$stats[1]
maxVal <- boxplot.stats(dataset$HDL)$stats [5]
myValue <- mean(dataset$HDL)
dataset [dataset$HDL < minVal | dataset$HDL > maxVal, "HDL"] <- myValue
boxplot(dataset$HDL)
boxplot.stats(dataset$HDL)$out

outliers <- boxplot(dataset$HDL, plot=FALSE)$out
dataset <- dataset[-which(dataset$HDL %in% outliers),]
boxplot(dataset$HDL)
boxplot.stats(dataset$HDL)$out ###after
########################################

#LDL
```

```
boxplot.stats(dataset$LDL)$out   ##before
minVal <- boxplot.stats(dataset$LDL)$stats[1]
maxVal <- boxplot.stats(dataset$LDL)$stats [5]
myValue <- mean(dataset$LDL)
dataset [dataset$LDL < minVal | dataset$LDL > maxVal, "LDL"] <- myValue
boxplot(dataset$LDL)
boxplot.stats(dataset$LDL)$out

outliers <- boxplot(dataset$LDL, plot=FALSE)$out
dataset <- dataset[-which(dataset$LDL %in% outliers),]
boxplot(dataset$LDL)
boxplot.stats(dataset$LDL)$out ###after
###############################################

#hemoglobin

boxplot.stats(dataset$hemoglobin)$out   ##before
minVal <- boxplot.stats(dataset$hemoglobin)$stats[1]
maxVal <- boxplot.stats(dataset$hemoglobin)$stats [5]
myValue <- mean(dataset$hemoglobin)
dataset [dataset$hemoglobin < minVal | dataset$hemoglobin > maxVal, "hemoglobin"] <-
myValue
boxplot(dataset$hemoglobin)
boxplot.stats(dataset$hemoglobin)$out

outliers <- boxplot(dataset$hemoglobin, plot=FALSE)$out
dataset <- dataset[-which(dataset$hemoglobin %in% outliers),]
boxplot(dataset$hemoglobin)
boxplot.stats(dataset$hemoglobin)$out ###after
#######################################

#Urine.protein

boxplot.stats(dataset$Urine.protein)$out   ##before
minVal <- boxplot.stats(dataset$Urine.protein)$stats[1]
maxVal <- boxplot.stats(dataset$Urine.protein)$stats [5]
myValue <- mean(dataset$Urine.protein)
dataset [dataset$Urine.protein < minVal | dataset$Urine.protein > maxVal, "Urine.protein"] <-
myValue
boxplot(dataset$Urine.protein)
boxplot.stats(dataset$Urine.protein)$out

outliers <- boxplot(dataset$Urine.protein, plot=FALSE)$out
dataset <- dataset[-which(dataset$Urine.protein %in% outliers),]
boxplot(dataset$Urine.protein)
boxplot.stats(dataset$Urine.protein)$out ###after
```

```
##########################################

#serum.creatinine

boxplot.stats(dataset$serum.creatinine)$out   ##before
minVal <- boxplot.stats(dataset$serum.creatinine)$stats[1]
maxVal <- boxplot.stats(dataset$serum.creatinine)$stats [5]
myValue <- mean(dataset$serum.creatinine)
dataset [dataset$serum.creatinine < minVal | dataset$serum.creatinine > maxVal,
"serum.creatinine"] <- myValue
boxplot(dataset$serum.creatinine)
boxplot.stats(dataset$serum.creatinine)$out
###################################################

#AST

boxplot.stats(dataset$AST)$out   ##before
minVal <- boxplot.stats(dataset$AST)$stats[1]
maxVal <- boxplot.stats(dataset$AST)$stats [5]
myValue <- mean(dataset$AST)
dataset [dataset$AST < minVal | dataset$AST > maxVal, "AST"] <- myValue
boxplot(dataset$AST)
boxplot.stats(dataset$AST)$out

outliers <- boxplot(dataset$AST, plot=FALSE)$out
dataset <- dataset[-which(dataset$AST %in% outliers),]
boxplot(dataset$AST)
boxplot.stats(dataset$AST)$out ###after
########################################################

#ALT

boxplot.stats(dataset$ALT)$out   ##before
minVal <- boxplot.stats(dataset$ALT)$stats[1]
maxVal <- boxplot.stats(dataset$ALT)$stats [5]
myValue <- mean(dataset$ALT)
dataset [dataset$ALT < minVal | dataset$ALT > maxVal, "ALT"] <- myValue
boxplot(dataset$ALT)
boxplot.stats(dataset$ALT)$out

outliers <- boxplot(dataset$ALT, plot=FALSE)$out
dataset <- dataset[-which(dataset$ALT %in% outliers),]
boxplot(dataset$ALT)
boxplot.stats(dataset$ALT)$out ###after
###########################################
```

```
#Gtp

boxplot.stats(dataset$Gtp)$out   ##before
minVal <- boxplot.stats(dataset$Gtp)$stats[1]
maxVal <- boxplot.stats(dataset$Gtp)$stats [5]
myValue <- mean(dataset$Gtp)
dataset [dataset$Gtp < minVal | dataset$Gtp > maxVal, "ALT"] <- myValue
boxplot(dataset$Gtp)
boxplot.stats(dataset$Gtp)$out

outliers <- boxplot(dataset$Gtp, plot=FALSE)$out
dataset <- dataset[-which(dataset$Gtp %in% outliers),]
boxplot(dataset$Gtp)
boxplot.stats(dataset$Gtp)$out

#Correlation analysis

cor(dataset$weight.kg.,dataset$hemoglobin)
cor(dataset$weight.kg.,dataset$Gtp)
cor(dataset$weight.kg.,dataset$Cholesterol)
cor(dataset$weight.kg.,dataset$HDL)
cor(dataset$weight.kg.,dataset$AST)
cor(dataset$weight.kg.,dataset$Urine.protein)
#Discretization
install.packages("arules")
library(arules)
x <- dataset[,2]
table(arules::discretize(x, breaks = 3))
```

- Data mining tasks:

classification :

```
##1-Split data (70% - 30%) #MyFourmula and tables:
set.seed(1234)
ind <- sample(2,nrow(data),replace=TRUE,prob = c(0.7,0.3))
trainData <-data[ind==1,]
testData <-data[ind==2,]
myFormula <- smoking ~ gender     +age+  height.cm.     +weight.kg.+   waist.cm.
        +eyesight.left.+        eyesight.right.+        hearing.left.+  hearing.right.+systolic
        +relaxation+   fasting.blood.sugar+   Cholesterol+   triglyceride+   HDL+  LDL+
        hemoglobin+   Urine.protein+serum.creatinine+      AST+   ALT    +Gtp+  dental.caries+
        tartar
dataset_ctree <-ctree(myFormula, data=trainData)
table(predict(dataset_ctree), trainData$smoking)
#Trees:
print(dataset_ctree)
```

```
plot(dataset_ctree)
plot(dataset_ctree, type ="simple" )

##1-Split data (80% - 20%) #MyFourmula and tables:
set.seed(1234)
ind <- sample(2,nrow(data),replace=TRUE,prob = c(0.8,0.2))
trainData <-data[ind==1,]
testData <-data[ind==2,]
myFormula <- smoking ~ gender      +age+  height.cm.      +weight.kg.+   waist.cm.
        +eyesight.left.+          eyesight.right.+          hearing.left.+  hearing.right.+systolic
        +relaxation+   fasting.blood.sugar+   Cholesterol+   triglyceride+   HDL+  LDL+
        hemoglobin+   Urine.protein+serum.creatinine+       AST+  ALT    +Gtp+  dental.caries+
        tartar
dataset_ctree <-ctree(myFormula, data=trainData)
table(predict(dataset_ctree), trainData$smoking)
#Trees:
print(dataset_ctree)
plot(dataset_ctree)
plot(dataset_ctree, type ="simple" )

##1-Split data (60% - 40%) #MyFourmula and tables:
set.seed(1234)
ind <- sample(2,nrow(data),replace=TRUE,prob = c(0.6,0.4))
trainData <-data[ind==1,]
testData <-data[ind==2,]
myFormula <- smoking ~ gender      +age+  height.cm.      +weight.kg.+   waist.cm.
        +eyesight.left.+          eyesight.right.+          hearing.left.+  hearing.right.+systolic
        +relaxation+   fasting.blood.sugar+   Cholesterol+   triglyceride+   HDL+  LDL+
        hemoglobin+   Urine.protein+serum.creatinine+       AST+  ALT    +Gtp+  dental.caries+
        tartar
dataset_ctree <-ctree(myFormula, data=trainData)
table(predict(dataset_ctree), trainData$smoking)
#Trees:
print(dataset_ctree)
plot(dataset_ctree)
plot(dataset_ctree, type ="simple" )
clustering :
dataset$gender<-as.numeric(dataset$gender)
dataset$tartar<-as.numeric(dataset$tartar)
dataset$oral<-as.numeric(dataset$oral)
#after scaleing there is an attribute with null values in all the row so we have to remove it
dataset2 <- dataset[,-23]
summary(dataset2)
str(dataset2)
# k-means clustering set a seed for random number generation  to make the results reproducible
set.seed(9000)
```

```
# prepreocessing
#Data types should be transformed into numeric types before clustering.
dataset2 <- scale(dataset2)
View(dataset2)
```

- evaluate:

```
classification:
#Test:
testPred <- predict(dataset_ctree, newdata=testData)
##Evalute Model:
table(testPred, testData$smoking)
coMa <-confusionMatrix(testPred,testData$smoking)
 acc <-(coMa$overallU["Accuracy"]*100 )
 print(acc)
 print(coMa)
 precision(testPred,testData$smoking)
clustering :
# run kmeans clustering to find 5 clusters
kmeans.result <- kmeans(dataset2, 5)
# print the clusterng result
kmeans.result
 ###Cluster Validation
library(cluster)
#average for each cluster
avg_sil <- silhouette(kmeans.result$cluster,dist(dataset2))
fviz_silhouette(avg_sil)#k-means clustering with estimating k and initializations
# run kmeans clustering to find 2 clusters
kmeans.result <- kmeans(dataset2, 2)
# print the clusterng result
kmeans.result
## visualize clustering
#install.packages("factoextra")
library(factoextra)
fviz_cluster(kmeans.result, data = dataset2)
###Cluster Validation
library(cluster)
#average for each cluster
avg_sil <- silhouette(kmeans.result$cluster,dist(dataset2))
fviz_silhouette(avg_sil)#k-means clustering with estimating k and initializations


#####################################################################################
# run kmeans clustering to find 7 clusters
kmeans.result <- kmeans(dataset2, 7)
# print the clusterng result
```

```
kmeans.result
## visualize clustering
#install.packages("factoextra")
library(factoextra)
fviz_cluster(kmeans.result, data = dataset2)
# plot cluster points
plot(dataset2[, c("smoking","systolic")], col = (kmeans.result$cluster) )
# plot cluster centers
points(kmeans.result$centers[, c("smoking","systolic")],  col = 1:4, pch = 8, cex=2)

###Cluster Validation
library(cluster)
#average for each cluster
avg_sil <- silhouette(kmeans.result$cluster,dist(dataset2))
fviz_silhouette(avg_sil)#k-means clustering with estimating k and initializations
#############################################################################
###############

library(NbClust)
#a)fviz_nbclust() with silhouette method using library(factoextra)
fviz_nbclust(dataset2, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
#b) NbClust validation
fres.nbclust <- NbClust(dataset2, distance="euclidean", min.nc = 2, max.nc = 10, 
method="kmeans", index="all")
```

10 References

[1] A. Baku, "Kaggle," Kaggle, 1 December 2022. [Online]. Available: https://www.kaggle.com/code/eisgandar/smoking-signal-of-body-classification/notebook. [Accessed 1 January 2023].

[2] S. kukuroo3, "Kaggle," 15 May 2022. [Online]. Available: https://www.kaggle.com/datasets/kukuroo3/body-signal-of-smoking. [Accessed 1 January 2023].

[3] A. Polonioli, "AI Search Blog," Coveo, 30 January 2023. [Online]. Available: https://www.coveo.com/blog/clustering-and-classification-in-ecommerce/. [Accessed 31 January 2023].

[4] J. Han, M. Micheline and J. Pei, "Data Mining: Concepts and techniques 3rd edition," in *Data Mining: Concepts and techniques 3rd edition*, Elsevier Science, Elsevier Science, 2011, pp. 83-117.

11 Tasks Distribution

ID	Name	Responsibilities
442201381	Sarah k Jwuied	Task was divided equally
442202526	Nouf Saleh Aldakheel	
442200304	Basma Alamoud	