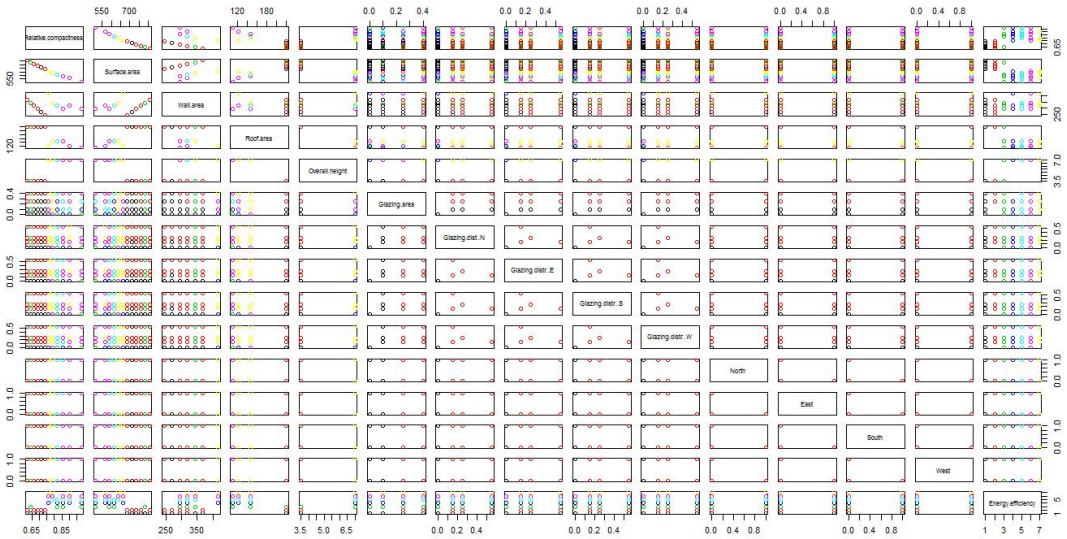
Report Classification and Clustering Lab Session

WANG Andi

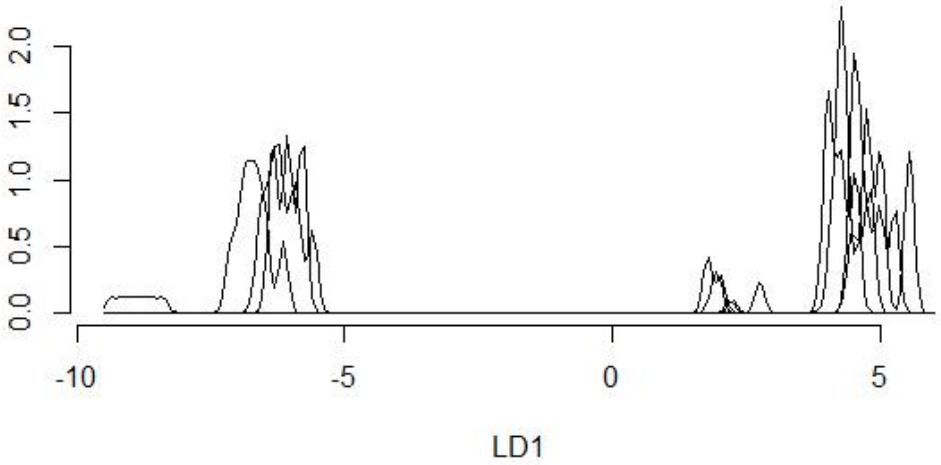
Exploratory analysis :

Using *pairs* :

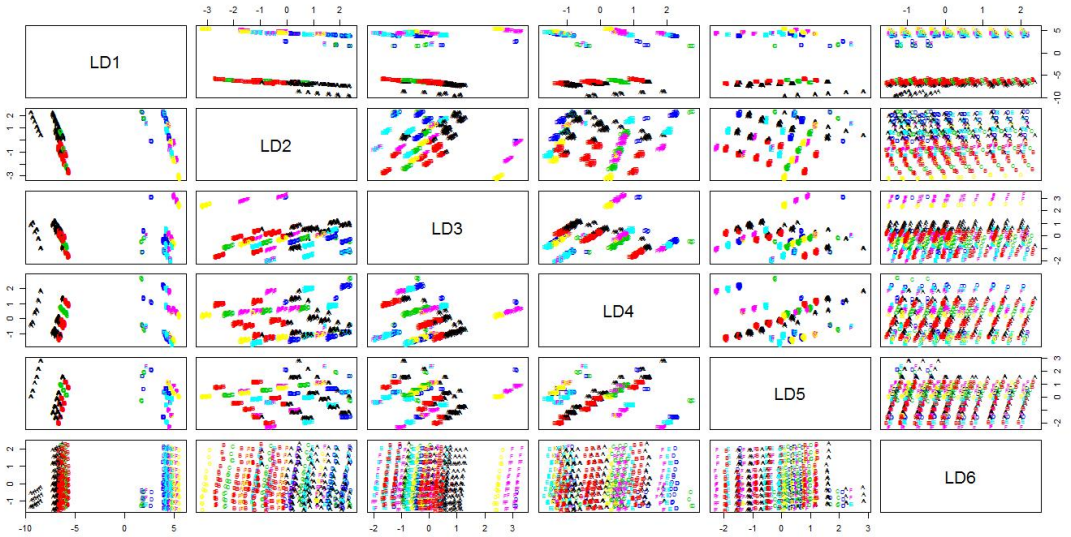


Lda model :

When plot the model in dim=1 and type="density":



Then we plot all the linear discriminate:



We can see from this image: this method of classification does not work well. The boundary of each class is not clear, and does not well corresponds with the data given. And then we analyze the matrix of confusion:

```
> table(ldapred$class,data[,15])#confusion

     A   B   C   D   E   F   G
A  151   5  20   0   0   0   0
B    4 124  40   0   0   0   0
C   20  20   0   0   0   0   0
D    0   0  10  56  32  12  10
E    0   0   0  18  51  11   0
F    0   0   0   4   0  95   5
G    0   0   0   0  20   0  60
```

By this matrix, we can confirm what we said. Many points are confused when comparing with the datas given.

Logistic regression:

matrix of confusion:

```
RL.pred   A   B   C   D   E   F   G
      A 150   6  20   0   0   0   0
      B   8 127  28   0   0   0   0
      C  17  16  18   2   0   0   0
      D   0   0   4  54  32   2   0
      E   0   0   0  18  51  11   0
      F   0   0   0   4   0 103  15
      G   0   0   0   0  20   2  60
```

By this matrix, we can all so find that so many points are confused among two or more classes.

Kmeans model:

After we make the model with the datas and class them into 7 groups, we make a matrix de confusion of the groups and the data original:

K=7:

```
> table(cl7$cluster,data[,15])

      A   B   C   D   E   F   G
  1   0   0   4  38  22   0   0
  2   0   0   4  20  20  20   0
  3   0   0   0   4   0  40  20
  4   0   0   2  16  59  46   5
  5  52  80  60   0   0   0   0
  6   0   0   0   0   2  12  50
  7 123  69   0   0   0   0   0
```

K=2:

```
> table(cl2$cluster,data[,15])

      A   B   C   D   E   F   G
  1 175 149  60   0   0   0   0
  2   0   0  10  78 103 118  75
```

In this matrix we can see the same problem of separation, the boundary is not clear and many point are confused.

Clustering on the scaled data

It is important to scale the data before clustering. The issue is what represents a good measure of distance between cases. This step is so important for the clustering. Because it corresponds with the calculation of the distance. And for all the clustering, there will need to calculate the distance.

Kmeans model:

This model uses the Euclidian distance. We apply the model onto the scaled data, and then we get the matrix confusion:

```
> table(mkmeans$cluster,data[,15])

    A  B  C  D  E  F  G
  1 21 24  9  0  0  0  0
  2  0  0 10 60 81 52  5
  3 36 30 12  0  0  0  0
  4  0  0  0 14 20 24 14
  5  0  0  0  4  2 42 56
  6 90 63 27  0  0  0  0
  7 28 32 12  0  0  0  0
```

We can see we could not well cluster the data into 7 clusters. Many points are still confused.

We also test k=2, k=3,k=9:

K=2:

```
> table(mkmeans2$cluster,data[,15])
```

```
     A   B   C   D   E   F   G
1    0   0  10  78 103 118  75
2  175 149  60   0   0   0   0
```

K=3:

```
> table(mkmeans3$cluster,data[,15])

     A   B   C   D   E   F   G
1    0   0   2  19  26  29  20
2    0   0   8  59  77  89  55
3  175 149  60   0   0   0   0
```

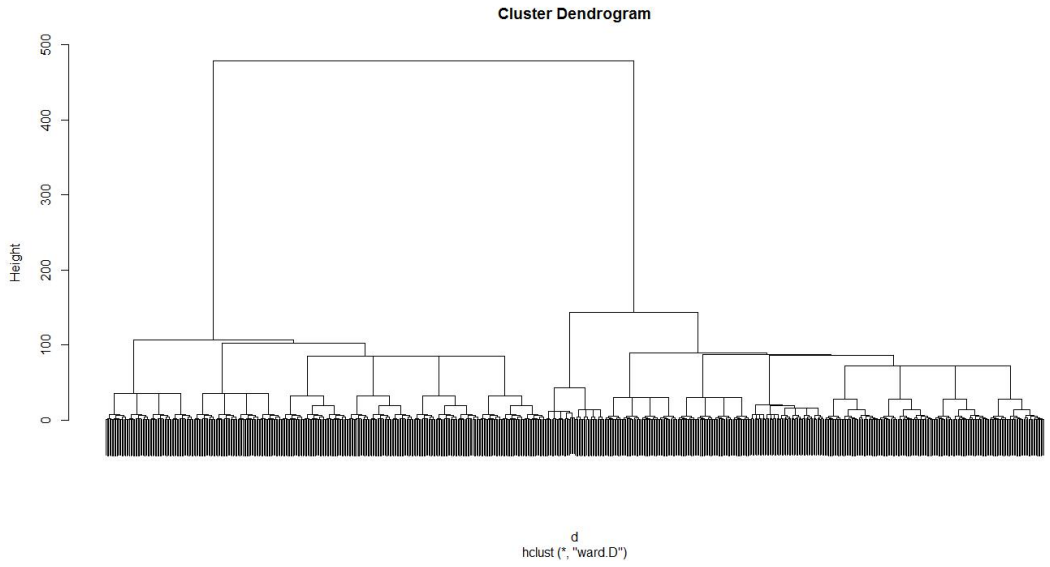
K=9:

```
> table(mkmeans9$cluster,data[,15])

    A  B  C  D  E  F  G
1   0  0  3 13 16 17 11
2   0  0  0 14 19 24 15
3   0  0  0 11 15 18 10
4   0  0  3 12 16 17 12
5  43 38 15  0  0  0  0
6   0  0  2 16 21 24 15
7   0  0  2 12 16 18 12
8  44 37 15  0  0  0  0
9  88 74 30  0  0  0  0
```

We can see when k=2, the data can be almost separated into 2 groups , one for (A B C) and one for (D E F G).

Hierarchies:

When using distance Euclidean:



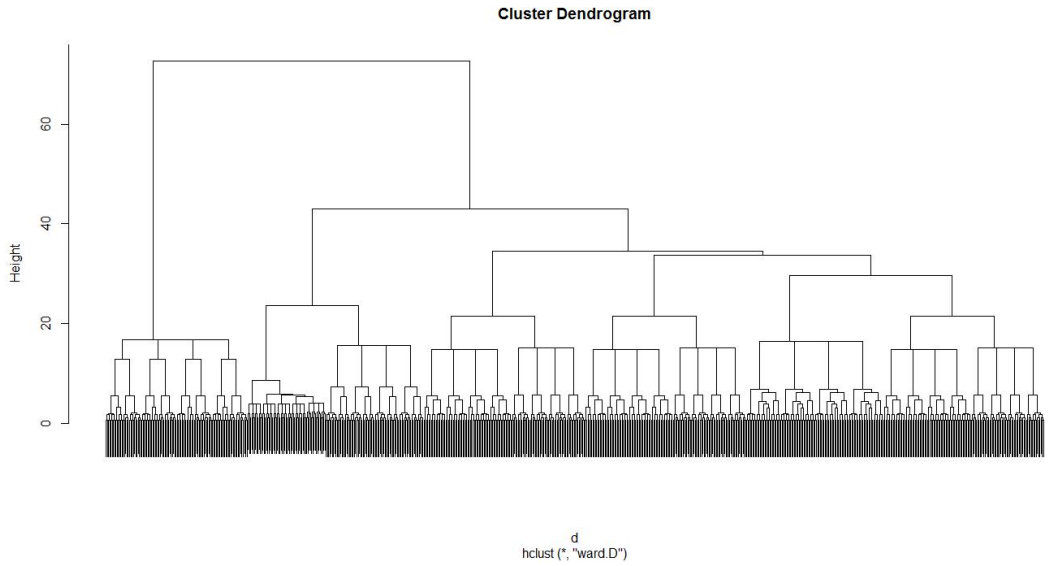
When cut into 2 groups:

```
> table(cu,data[,15])

cu     A    B    C    D    E    F    G
  1   24    0   10   78  103  118   75
  2  151  149   60    0    0    0    0
```

When into 3:

```
> table(cu,data[,15])

cu     A    B    C    D    E    F    G
  1   24    0   10   10    2    2    0
  2    0    0    0   68  101  116   75
  3  151  149   60    0    0    0    0
```

When using distance maximum:



This model has many points confusing than the others when separated into 2 or 3 , we will not discuss it in future:

```
> table(cu,data[,15])

cu     A    B    C    D    E    F    G
  1   47    9   10   18   18    8    6
  2  128  140   60   60   85  110   69
> cu<-cutree(hc1,3) #test 2 3 7
> table(cu,data[,15])

cu     A    B    C    D    E    F    G
  1   47    9   10   18   18    8    6
  2    0    0    0   18   25   64   37
  3  128  140   60   42   60   46   32
```

When using distance manhattan:

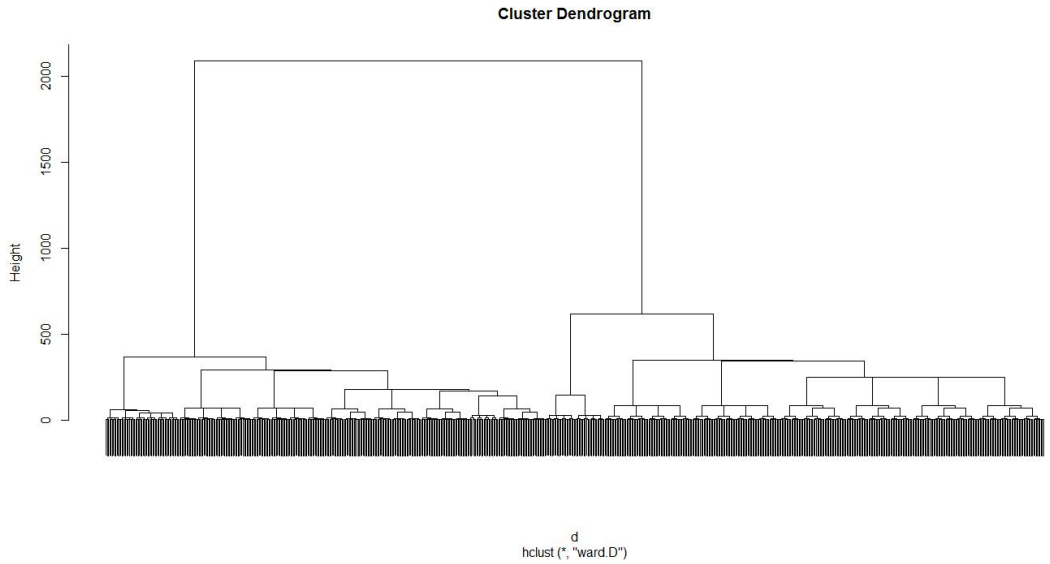


Table of confusion:

When cut into 2 groups:

```
> table(cu,data[,15])
```

```
cu     A    B    C    D    E    F    G
  1 175  149   70   10    2    2    0
  2   0    0    0   68  101  116   75
```

When cut into 3:

```
> table(cu,data[,15])

cu     A    B    C    D    E    F    G
  1   24    0   10   10    2    2    0
  2    0    0    0   68  101  116   75
  3  151  149   60    0    0    0    0
```

We can see when we use distance de manhattan for this model, we can separate into 2 clusters with minimum points confusing

K-Medoids:

K=2:

```
> table(m2$clustering,data[,15])

      A    B    C    D    E    F    G
  1   32   16   10   77  102  105   58
  2  143  133   60    1    1   13   17
```

K=3:

```
> table(m3$clustering,data[,15])

      A    B    C    D    E    F    G
  1   0    0    8   57   91   92   40
  2 106   71   16   11    6   13   17
```

```
  3  69  78  46  10   6  13  18
```

K=7:

```
> table(m7$clustering,data[,15])

    A  B  C  D  E  F  G
1   0  0  2 23 36 43 24
2   0  0  4 26 36 42 20
3   0  0  3 20 25 30 18
4  43 38 16  9  6  3 13
5  45 36 15  0  0  0  0
6  43 38 15  0  0  0  0
7  44 37 15  0  0  0  0
```

Knn method:

K=1:

```
mknn1  A  B  C  D  E  F  G
    A 39 30  0  0  0  0  0
    B 18 26  0  0  0  0  0
    C  1  0 19  0  0  0  0
    D  0  0  3  9 14  4  0
    E  0  0  0 10 10  7  1
    F  0  0  0  4  9 19  9
    G  0  0  0  0  1 14  9
```

K=3:

```
mknn3  A  B  C  D  E  F  G
    A 43 20  0  0  0  0  0
    B 14 36  0  0  0  0  0
    C  1  0 19  0  0  0  0
    D  0  0  3  9 11  4  0
    E  0  0  0 10 12  4  1
    F  0  0  0  4 10 26  4
    G  0  0  0  0  1 10 14
```

We do also the test of k=7 and 11, the matrix are similar. We can see that the model can separate the data into 2 groups one for (A, B, C) and one for the others with just a few points confusing.

And also use cross-validation for k=5:

```
model   A   B   C  D  E  F  G
    A 107  45   0  0  0  0  0
    B  64 104   0  0  0  0  0
    C   4   0  60  0  0  0  0
    D   0   0   9 26 24  9  0
    E   0   0   1 38 50 25  1
    F   0   0   0 14 27 71 20
    G   0   0   0  0  2 13 54
```

We can see by this model, even there are also points confusing ,and can not separate the data into 7, we can almost separate the data into 2 group.

When we compare these methods for classification, we can find that the method k-medoids , performs very poorly, even in the separation into 2 groups, it can not work well. There are more points confusing .

Classification:

I will chose knn, kmeans and lda, regression logistic:

Lda:

```
> table(predlda$class,test[,15])

    A  B  C  D  E  F  G
A  55  6  5  0  0  0  0
B   2 43 13  0  0  0  0
C   5  3  1  0  0  0  0
D   0  0  3 13 12  5  3
E   0  0  0 16 16  7  0
F   0  0  0  3  0 22  4
G   0  0  0  0  4  0 15
```

3/256=1,17% for 2 groups

Classrate=98.83%

(A B C) (D E)(F G) 3 groups

Classrate=1-(3+5+3+7+3+4)/256=90.23%

Regression logistic:

```
> table(RL.pred,test[,15])

RL.pred  A  B  C  D  E  F  G
      A 56  3  5  0  0  0  0
      B  2 35  7  0  0  0  0
      C  4 14 10  4  2  0  0
      D  0  0  0 14 10  3  3
      E  0  0  0 13 14  2  0
      F  0  0  0  1  3 27  3
      G  0  0  0  0  3  2 16
```

```
>
```

6/256=3.34%

Classrate=96.66% for 2 groups

(A B C)(D E)(F G) for 3 groups:

Classrate=1-(4+2+3+3+2+1+3+3)/256=91.80%

Knn:

K=3:

---

mknn3	A	B	C	D	E	F	G
A	43	20	0	0	0	0	0
B	14	36	0	0	0	0	0
C	1	0	19	0	0	0	0
D	0	0	3	9	11	4	0
E	0	0	0	10	12	4	1
F	0	0	0	4	10	26	4
G	0	0	0	0	1	10	14

For this model, we have already separate into 2 in the former tests, and

Classrate=1-3/256=98.83%

Into 3 groups:

1-(3+4+4+1+4+10+1)/256=89.45%

Kmeans:

Because the function kmeans is the function internal, so we can directly use it for the test:

```
> table(pre$cluster,test[,15])
     A  B  C  D  E  F  G
  1  0  0  3 32 32 34 22
  2 62 52 19  0  0  0  0
```

Classrate=1-3/256=98.83%

(b)we can still calculate by the methods linear, but the non-linear performs much more poor. As the clustering needs to calculate the distance and the raw data will influence the calculation of the distance

(c)

We make a new data with 1st colonne and 11th colonnes.