H-1B AND PERMANENT VISA APPLICATION CASES ANALYSIS

Data Mining Report (Part II)

ABSTRACT

Using different clustering methods to deeply analyze the data of U.S. Permanent visa.

Hua, Mengli/ Ma, Liang

CSE7331 Data Mining

Contents

III. Data Preparation

We choose the job_info_work_state and country_of_citizenship from the dataset "US Permanent Visa" as the features which will be clustering for further analysis. There are several reasons that lead us to make this choice.

First of all, it is obvious that either the class of admission or the country of citizenship not only has a large range of values but also share similarities between its values. Secondly, by clustering, it becomes more visible for us to figure out what kind of visa holders (matches other conditionings) more possibly get a certified for their green card applications. This will be helpful when using extra information to achieve deep understanding. Finally, for the country_of_citizenship, we need far more relative information from the datasets. Clustering is usable as a further analyzing method under this condition.

Thus, we will use job_info_work_state and case_status for the clustering of class_of_admission; use case_status and pw_amount_9089 for country_of_citizenship' s clustering. The scale of measurement of the features are chosen based on the business analysis. We will use Euclidean Distance to calculate the distance between objects, because that Euclidean distance can reflect the absolute difference in individual numerical characteristics. New objects do not affect the distance between any two variables. And the metrics of the object attributes are the same (both percentages) in this project, the result has minor impact.

IV. Modeling

In this part, we will describe each step of the clustering and the codes for implementation. We will do K-means on the country_of_citizenship while do hierarchy on class_of_admission.

1. Hierarchy clustering on class_of_admission

First and foremost, it is important to processing the data we need. So "subset()" function has been used to get the specific part of data we need. After that, we do clearing (delete the whole row) for all "NA" values via "na.omit()" function. Cause the state is not as clean as we need, we have to clean it again. This time we use "droplevels()" to omit the useless abbreviation of states and visa types. Coding of this part is as following:

```
> usperm3<-subset(usperm, select = c(case_status,class_of_admission,job_info_work_state))
```

```
> usperm3<-na.omit(usperm3)
> summary(usperm3)
          case_status         class_of_admission job_info_work_state
 Certified        :181933   H-1B      :283019   CALIFORNIA: 47698
 Certified-Expired:148586             : 22845   CA        : 39620
 Denied           : 25649   L-1       : 19938   TEXAS     : 24872
 Withdrawn        : 18194   F-1       : 14946   TX        : 21230
                            Not in USA:  8588   NEW JERSEY: 16726
                            TN        :  4265   NEW YORK  : 16089
                            (Other)   : 20761   (Other)   :208127
```

Then, we start to transfer the frequencies of each case status into percentage.

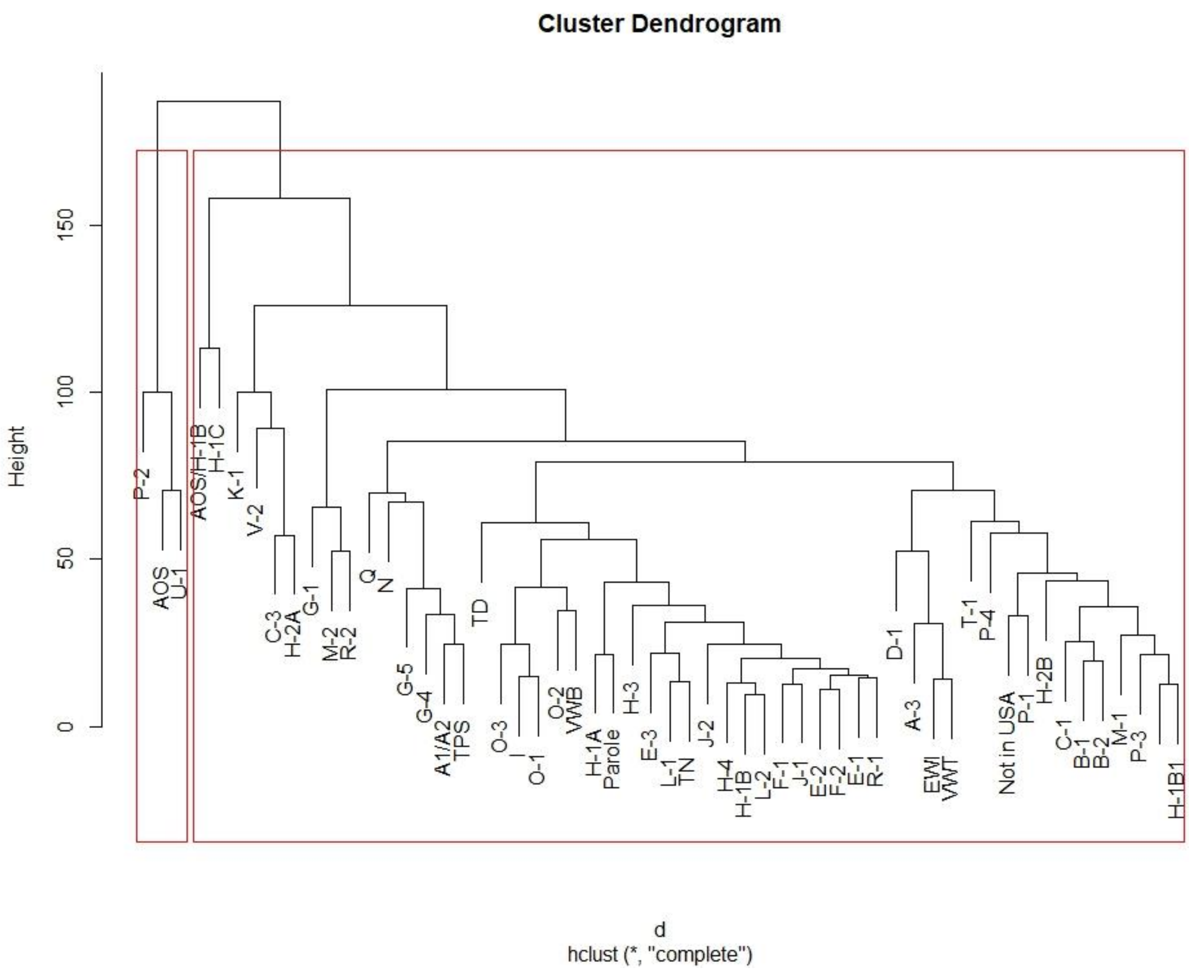
```
> agg_case <- as.data.frame.matrix(tbl_case)
> agg_case <- agg_case / rowSums(agg_case) * 100
> head(tbl_case)

          Certified Certified-Expired Denied Withdrawn
  A-3             7                 7     14         1
  A1/A2          66                52     31         6
  AOS             0                 1      0         0
  AOS/H-1B        2                 0      0         0
  B-1           216               129    245        29
  B-2          1316               949    896       169
> head(agg_case)
         Certified Certified-Expired   Denied Withdrawn
A-3       24.13793          24.13793 48.27586  3.448276
A1/A2     42.58065          33.54839 20.00000  3.870968
AOS        0.00000         100.00000  0.00000  0.000000
AOS/H-1B 100.00000           0.00000  0.00000  0.000000
B-1       34.89499          20.84006 39.57997  4.684976
B-2       39.51952          28.49850 26.90691  5.075075
```

Same for the state data. Use “str()” we can see the details and it is suitable for us to do combining next.

```
> str(tbl_state)
 'table' int [1:56, 1:57] 587 0 0 0 0 4 11 0 0 0 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:56] "" "A-3" "A1/A2" "AOS" ...
  ..$ : chr [1:57] "ALABAMA" "ALASKA" "ARIZONA" "ARKANSAS" ...
```

```
> head(aggh)
         Certified Certified-Expired   Denied Withdrawn   ALABAMA     ALASKA   ARIZONA  ARKANSAS
          44.31264          28.74654 19.22587  7.714951 2.5789728 0.04832828 0.5491850 1.7266377
A-3       24.13793          24.13793 48.27586  3.448276 0.0000000 0.00000000 0.0000000 0.0000000
A1/A2     42.58065          33.54839 20.00000  3.870968 0.0000000 0.00000000 0.0000000 0.0000000
AOS        0.00000         100.00000  0.00000  0.000000 0.0000000 0.00000000 0.0000000 0.0000000
AOS/H-1B 100.00000           0.00000  0.00000  0.000000 0.0000000 0.00000000 0.0000000 0.0000000
B-1       34.89499          20.84006 39.57997  4.684976 0.6462036 0.16155089 0.1615509 0.3231018
         CALIFORNIA   COLORADO CONNECTICUT  DELAWARE DISTRICT OF COLUMBIA
           16.20315  0.6414481   0.7468916 0.4085936            0.6765959
A-3        17.24138  0.0000000   6.8965517 0.0000000           17.2413793
A1/A2      14.19355  1.9354839   0.0000000 0.0000000           14.8387097
AOS       100.00000  0.0000000   0.0000000 0.0000000            0.0000000
AOS/H-1B    0.00000 50.0000000   0.0000000 0.0000000            0.0000000
B-1        21.64782  0.6462036   0.9693053 0.0000000            0.9693053
         FEDERATED STATES OF MICRONESIA   FLORIDA  GEORGIA       GUAM   HAWAII      IDAHO
                                      0  5.671983 6.440842 0.06150872 0.101050 0.05272176
A-3                                   0  3.448276 0.000000 0.00000000 0.000000 0.00000000
A1/A2                                 0  3.225806 3.225806 0.00000000 1.290323 0.00000000
AOS                                   0  0.000000 0.000000 0.00000000 0.000000 0.00000000
AOS/H-1B                              0  0.000000 0.000000 0.00000000 0.000000 0.00000000
B-1                                   0 11.954766 2.423263 0.00000000 0.000000 0.16155089
          ILLINOIS   INDIANA      IOWA    KANSAS  KENTUCKY LOUISIANA     MAINE MARSHALL ISLANDS
         1.5201441 0.3382980 0.3339045 0.3382980 0.1889196 0.4744958 0.3382980       0.00439348
A-3      0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000       0.00000000
A1/A2    0.6451613 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000       0.00000000
AOS      0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000       0.00000000
AOS/H-1B 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000       0.00000000
B-1      5.3311793 0.4846527 0.1615509 0.1615509 0.4846527 0.6462036 0.3231018       0.00000000
          MARYLAND MASSACHUSETTS  MICHIGAN MINNESOTA MISSISSIPPI  MISSOURI    MONTANA  NEBRASKA
          1.818901     1.3092571 2.3109705 0.8127938    1.247748 0.7644655 0.07908264 0.0878696
A-3      10.344828     0.0000000 0.0000000 0.0000000    0.000000 0.0000000 0.00000000 0.0000000
A1/A2    21.290323     0.6451613 0.6451613 0.6451613    0.000000 0.0000000 0.00000000 0.0000000
AOS       0.000000     0.0000000 0.0000000 0.0000000    0.000000 0.0000000 0.00000000 0.0000000
```

Next step is to calculate the distance and generate the hierarchy cluster plot.




```
> hclustd<-hclust(d)
> rect.hclust(hclustd,k=2)
> out.id=cutree(hclustd,k=2)
> out.id
                   A-3      A1/A2        AOS   AOS/H-1B        B-1        B-2        C-1        C-3 
         1           1          1          2          1          1          1          1          1 
       D-1         E-1        E-2        E-3        EWI        F-1        F-2        G-1        G-4 
         1           1          1          1          1          1          1          1          1 
       G-5        H-1A       H-1B      H-1B1       H-1C       H-2A       H-2B        H-3        H-4 
         1           1          1          1          1          1          1          1          1 
         I         J-1        J-2        K-1        L-1        L-2        M-1        M-2          N 
         1           1          1          1          1          1          1          1          1 
Not in USA         O-1        O-2        O-3        P-1        P-2        P-3        P-4          Q 
         1           1          1          1          1          2          1          1          1 
       R-1         R-2        T-1         TD         TN        TPS        U-1        V-2        VWB 
         1           1          1          1          1          1          2          1          1 
       VWT      Parole 
         1           1 
```

2. K-means clustering on country_of_citizenship

First and foremost, it is important to processing the data we need. So we use “subset()” function to get the specific part of data we need. Then use the “unite()” function based on the “tidyr” package to unit data in “country_of_citizenship” and “country_of_citzenship” (cause they are the same thing but separate into two columns) and then delete the wrong spelling one “country_of_citzenship”. The next step is to change the data type, use the “as.factor()” function. After that, we do clearing (delete the whole row) for all “NA” values via “na.omit()” function. The last step is to transfer all character factors into numeric ones and aggregate them. We do group as the same time by “aggregate(list())”. Coding results of this part is as following:

```
> userm1 <-subset(userm,select = c(case_status,country_of_citizenship,country_of_citzenship,pw_amount_9089))
> library(tidyr)
> userm1<-unite(userm1, "country_of_citizenship", country_of_citizenship, country_of_citzenship, sep = "", remove = FALSE)
> userm1$country_of_citizenship<-as.factor(userm1$country_of_citizenship)
> userm1<-subset(userm1,select=-3)
>
> summary(userm1)
            case_status      country_of_citizenship   pw_amount_9089
 Certified         :181933   INDIA       :205158     72,467.00:  2777
 Certified-Expired:148586   CHINA       : 28861     81765.0  :  2234
 Denied            : 25649   SOUTH KOREA: 24761              :  2216
 Withdrawn         : 18194   CANADA      : 14804     54,059.00:  1561
                              MEXICO      :  8961     76,378.00:  1393
                              PHILIPPINES:  8631     97219.0  :  1133
                              (Other)     : 83186     (Other)  :363048
```

```
>
> summary(userm1)
            case_status      country_of_citizenship pw_amount_9089
 Certified         :181876   INDIA       :204419   Min.    :       6
 Certified-Expired:148548   CHINA       : 28801   1st Qu.:   66435
 Denied            : 23523   SOUTH KOREA: 24675   Median :   85675
 Withdrawn         : 18181   CANADA      : 14726   Mean   :   85349
                              MEXICO      :  8602   3rd Qu.:  104396
                              PHILIPPINES:  8504   Max.   :13528320
                              (Other)     : 82401
```

```
> userm1$country_of_citizenship[which(userm1$country_of_citizenship=="")]<-NA
> userm1<-na.omit(userm1)
> agg_wage <-aggregate(userm1$pw_amount_9089, by = list(userm1$country_of_citizenship), FUN = function(x) median(x, na.rm = TRUE))
> head(agg_wage)
              Group.1       x
1         AFGHANISTAN 48963.0
2             ALBANIA 65540.5
3             ALGERIA 84520.5
4             ANDORRA 80995.0
5              ANGOLA 89169.5
6 ANTIGUA AND BARBUDA 73008.0
```

Next, we set the country as the row name. Then take the “country_of_citizenship” and “case_status” out of the data frame and put them into a table. Before transfer the table back into a data frame matrix, we need to see the details of the attributes in this table. The code of this section is showing as below:

```
> agg_wage <- data.frame(wage = agg_wage$x, row.names = agg_wage$Group.1)
> head(agg_wage)
                       wage
AFGHANISTAN         48963.0
ALBANIA             65540.5
ALGERIA             84520.5
ANDORRA             80995.0
ANGOLA              89169.5
ANTIGUA AND BARBUDA 73008.0
```

```
> head(tbl_status)

                      Certified Certified-Expired Denied Withdrawn
  AFGHANISTAN                17                 6      3         1
  ALBANIA                    68                53     13         8
  ALGERIA                    17                16      3         4
  ANDORRA                     0                 1      0         0
  ANGOLA                      8                 5      1         2
  ANTIGUA AND BARBUDA         7                 5      0         1
```

```
> str(tbl_status)
 'table' int [1:202, 1:4] 17 68 17 0 8 7 570 80 0 860 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:202] "AFGHANISTAN" "ALBANIA" "ALGERIA" "ANDORRA" ...
  ..$ : chr [1:4] "Certified" "Certified-Expired" "Denied" "Withdrawn"
> head(as.data.frame(tbl_status))
                 Var1      Var2 Freq
1         AFGHANISTAN Certified   17
2             ALBANIA Certified   68
3             ALGERIA Certified   17
4             ANDORRA Certified    0
5              ANGOLA Certified    8
6 ANTIGUA AND BARBUDA Certified    7
```

```
> agg_status <- as.data.frame.matrix(tbl_status)
> head(agg_status)
                    Certified Certified-Expired Denied Withdrawn
AFGHANISTAN                17                 6      3         1
ALBANIA                    68                53     13         8
ALGERIA                    17                16      3         4
ANDORRA                     0                 1      0         0
ANGOLA                      8                 5      1         2
ANTIGUA AND BARBUDA         7                 5      0         1
```

```
> str(agg_status)
'data.frame':   202 obs. of  4 variables:
 $ Certified        : int  17 68 17 0 8 7 570 80 0 860 ...
 $ Certified-Expired: int  6 53 16 1 5 5 432 59 0 657 ...
 $ Denied           : int  3 13 3 0 1 0 123 15 0 88 ...
 $ Withdrawn        : int  1 8 4 0 2 1 57 11 0 83 ...
> 
```

Now, we need to change the frequencies of each status into percentages.

```
> agg_status <- agg_status / rowSums(agg_status) * 100
> head(agg_status)
                    Certified Certified-Expired   Denied Withdrawn
AFGHANISTAN          62.96296          22.22222 11.11111  3.703704
ALBANIA              47.88732          37.32394  9.15493  5.633803
ALGERIA              42.50000          40.00000  7.50000 10.000000
ANDORRA               0.00000         100.00000  0.00000  0.000000
ANGOLA               50.00000          31.25000  6.25000 12.500000
ANTIGUA AND BARBUDA  53.84615          38.46154  0.00000  7.692308
```

It is time to combine the processed matrix back. During this operation, we meet a error like this:

```
> 
> agg <- cbind(agg_status, agg_wage)
Error in data.frame(..., check.names = FALSE) : 
  参数值意味着不同的行数: 202, 201
```

(cause the language of the operating system is Chinese)

After checking, we find out that we forgot to clean the “case_status”.

```
> agg_status$Certified[agg_status$Certified==0&agg_status$`Certified-Expired`==0&ag
g_status$Denied==0&agg_status$Withdrawn==0]
[1] NA
> agg_status<-na.omit(agg_status)
```

```
> agg <- cbind(agg_status, agg_wage)
> head(agg)
                    Certified Certified-Expired   Denied Withdrawn    wage
AFGHANISTAN          62.96296          22.22222 11.11111  3.703704 48963.0
ALBANIA              47.88732          37.32394  9.15493  5.633803 65540.5
ALGERIA              42.50000          40.00000  7.50000 10.000000 84520.5
ANDORRA               0.00000         100.00000  0.00000  0.000000 80995.0
ANGOLA               50.00000          31.25000  6.25000 12.500000 89169.5
ANTIGUA AND BARBUDA  53.84615          38.46154  0.00000  7.692308 73008.0
```

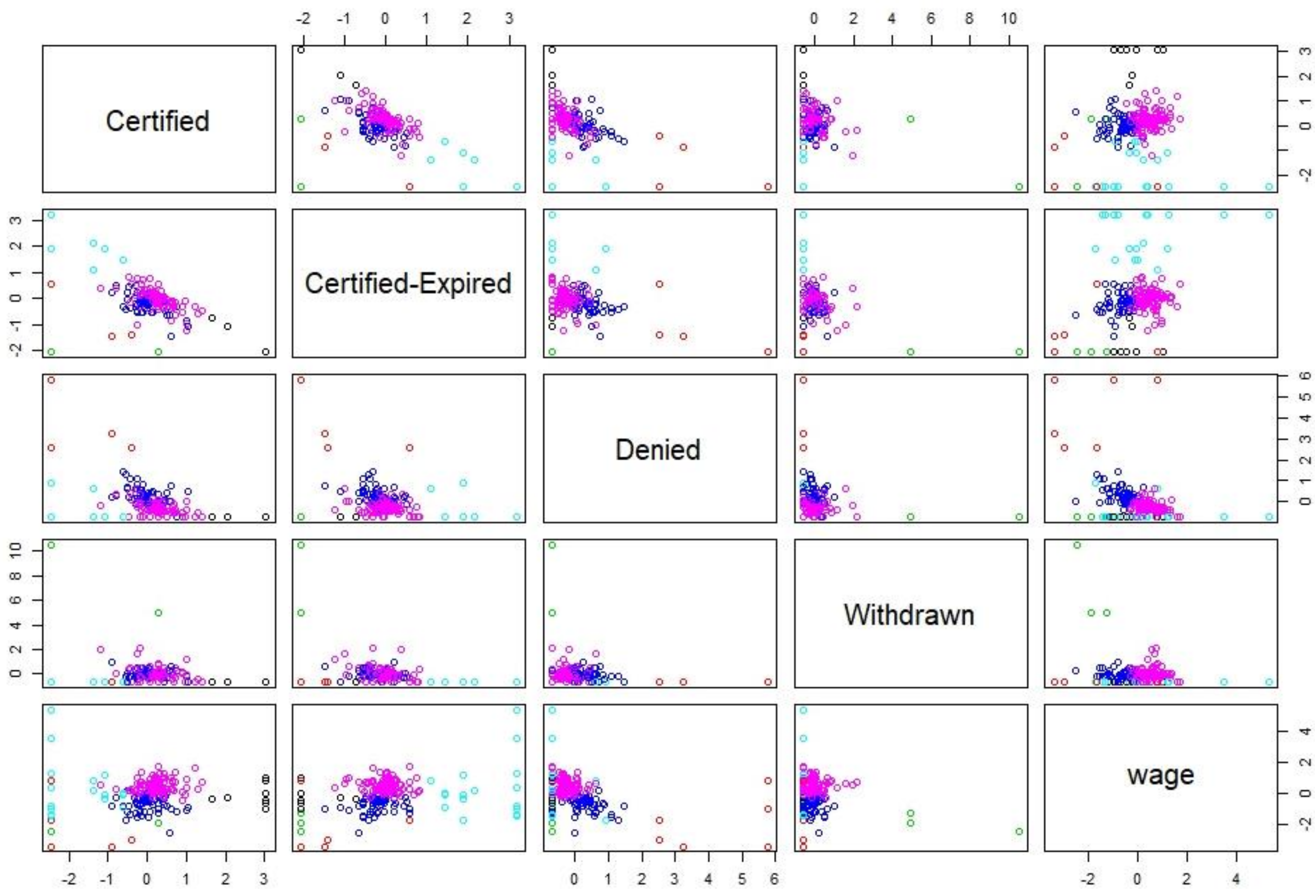
Problem has been solved. The next thing is to scale the data.

```
> agg_scaled <- scale(agg)
> head(agg_scaled)
                    Certified Certified-Expired      Denied  Withdrawn        wage
AFGHANISTAN         0.9978581       -0.87961810  0.02490913 -0.1860658 -1.09171602
ALBANIA             0.1674596       -0.09000150 -0.10163804  0.0279756 -0.32007841
ALGERIA            -0.1292858        0.04992020 -0.20869695  0.5121719  0.56338905
ANDORRA            -2.4702769        3.18711196 -0.69387882 -0.5967938  0.39928660
ANGOLA              0.2838303       -0.40758693 -0.28956060  0.7894133  0.77978738
ANTIGUA AND BARBUDA 0.4956847       -0.03052061 -0.69387882  0.2562567  0.02751344
```

Everything is done for clustering.

```
> km <- kmeans(agg_scaled, centers = 6)
> pairs(agg_scaled, col = km$cluster)
>
```

```
> km$centers
     Certified Certified-Expired     Denied   Withdrawn        wage
1  0.235910842       -0.02199788 -0.2629043  0.02238258  0.47376799
2  2.740619145       -1.75931255 -0.6938788 -0.59679383 -0.07648774
3 -0.003197078       -0.20090502  0.2875605 -0.06040728 -0.71327995
4 -1.812351309        2.43186209 -0.5321515 -0.59679383  0.31595146
5 -0.634205437       -2.04154097 -0.6938788  6.79631104 -1.83704666
6 -1.950368927       -1.49466656  4.4879955 -0.59679383 -2.11746603
```

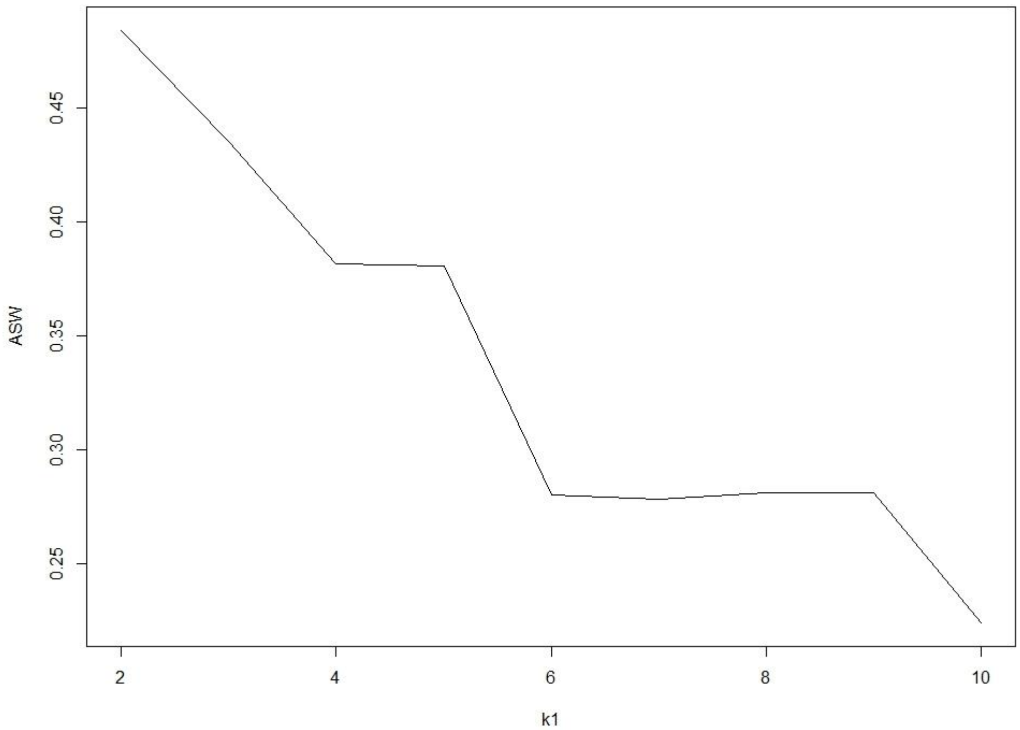


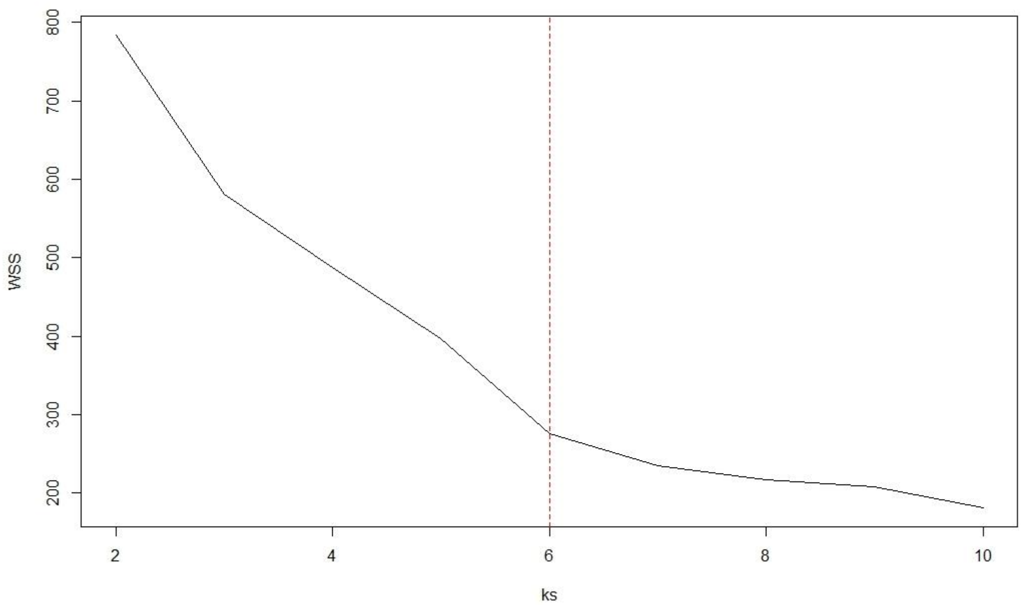
```
> sort(km$cluster)
            ALGERIA                      ANGOLA       ANTIGUA AND BARBUDA
                  1                           1                         1
          ARGENTINA                     ARMENIA                 AUSTRALIA
                  1                           1                         1
            AUSTRIA                  AZERBAIJAN                   BAHAMAS
                  1                           1                         1
            BAHRAIN                  BANGLADESH                   BELARUS
                  1                           1                         1
            BELGIUM                     BERMUDA                  BOTSWANA
                  1                           1                         1
             BRAZIL                    BULGARIA              BURKINA FASO
                  1                           1                         1
     BURMA (MYANMAR)                     CANADA                     CHILE
                  1                           1                         1
              CHINA                  COSTA RICA                   CROATIA
                  1                           1                         1
             CYPRUS              CZECH REPUBLIC                   DENMARK
                  1                           1                         1
              EGYPT                     ERITREA                   ESTONIA
                  1                           1                         1
           ETHIOPIA                     FINLAND                    FRANCE
                  1                           1                         1
              GABON                     GEORGIA                   GERMANY
                  1                           1                         1
              GHANA                      GREECE                 HONG KONG
                  1                           1                         1
            HUNGARY                     ICELAND                     INDIA
                  1                           1                         1
          INDONESIA                     IRELAND                    ISRAEL
                  1                           1                         1
              ITALY                 IVORY COAST                    JORDAN
                  1                           1                         1
         KAZAKHSTAN                      KUWAIT                KYRGYZSTAN
                  1                           1                         1
             LATVIA                     LEBANON                   LESOTHO
                  1                           1                         1
            LIBERIA                       LIBYA                 LITHUANIA
                  1                           1                         1
         LUXEMBOURG                       MACAU                MADAGASCAR
                  1                           1                         1
           MALAYSIA                   MAURITIUS                   MOROCCO
                  1                           1                         1
              NEPAL                 NETHERLANDS               NEW ZEALAND
                  1                           1                         1
            NIGERIA                      NORWAY                  PAKISTAN
                  1                           1                         1
          PALESTINE     PALESTINIAN TERRITORIES                    PANAMA
                  1                           1                         1
           PORTUGAL                     ROMANIA                    RUSSIA
                  1                           1                         1
             RWANDA                SAUDI ARABIA                   SENEGAL
                  1                           1                         1
             SERBIA       SERBIA AND MONTENEGRO              SIERRA LEONE
                  1                           1                         1
          SINGAPORE                    SLOVAKIA                  SLOVENIA
                  1                           1                         1
       SOUTH AFRICA                       SPAIN                 SRI LANKA
                  1                           1                         1
         ST VINCENT                       SUDAN                    SWEDEN
                  1                           1                         1
        SWITZERLAND                       SYRIA                    TAIWAN
                  1                           1                         1
           TANZANIA                        TOGO       TRINIDAD AND TOBAGO
                  1                           1                         1
            TUNISIA                      TURKEY  TURKS AND CAICOS ISLANDS
                  1                           1                         1
             UGANDA                     UKRAINE      UNITED ARAB EMIRATES
                  1                           1                         1
```

```
           UNITED KINGDOM     UNITED STATES OF AMERICA                      URUGUAY
                        1                            1                            1
               UZBEKISTAN                      VANUATU                    VENEZUELA
                        1                            1                            1
                    YEMEN                   YUGOSLAVIA                       ZAMBIA
                        1                            1                            1
                 ZIMBABWE                       BRUNEI                COTE d'IVOIRE
                        1                            2                            2
        EQUATORIAL GUINEA                GUINEA-BISSAU                     KIRIBATI
                        2                            2                            2
               MONTENEGRO                      NAMIBIA             PAPUA NEW GUINEA
                        2                            2                            2
              AFGHANISTAN                      ALBANIA                     BARBADOS
                        3                            3                            3
                   BELIZE                        BENIN                       BHUTAN
                        3                            3                            3
                  BOLIVIA       BOSNIA AND HERZEGOVINA       BRITISH VIRGIN ISLANDS
                        3                            3                            3
                 CAMBODIA                     CAMEROON                     COLOMBIA
                        3                            3                            3
DEMOCRATIC REPUBLIC OF CONGO                  DOMINICA           DOMINICAN REPUBLIC
                        3                            3                            3
                  ECUADOR                  EL SALVADOR                         FIJI
                        3                            3                            3
                  GRENADA                    GUATEMALA                       GUINEA
                        3                            3                            3
                   GUYANA                        HAITI                     HONDURAS
                        3                            3                            3
                     IRAN                         IRAQ                      JAMAICA
                        3                            3                            3
                    JAPAN                        KENYA                       KOSOVO
                        3                            3                            3
                MACEDONIA                       MALAWI                         MALI
                        3                            3                            3
         MARSHALL ISLANDS                       MEXICO                      MOLDOVA
                        3                            3                            3
                 MONGOLIA                    NICARAGUA                        NIGER
                        3                            3                            3
                 PARAGUAY                         PERU                  PHILIPPINES
                        3                            3                            3
                   POLAND SAINT VINCENT AND THE GRENADINES              SOUTH KOREA
                        3                            3                            3
              SOUTH SUDAN             ST KITTS AND NEVIS                     ST LUCIA
                        3                            3                            3
                 SURINAME                   TAJIKISTAN                     THAILAND
                        3                            3                            3
             TURKMENISTAN                      VIETNAM                      ANDORRA
                        3                            3                            4
                  BURUNDI               CAYMAN ISLANDS     CENTRAL AFRICAN REPUBLIC
                        4                            4                            4
                     CHAD                LIECHTENSTEIN                     MALDIVES
                        4                            4                            4
                    MALTA                   MAURITANIA                   MOZAMBIQUE
                        4                            4                            4
     NETHERLANDS ANTILLES                  NORTH KOREA                         OMAN
                        4                            4                            4
                    QATAR                   SEYCHELLES                      SOMALIA
                        4                            4                            4
             SOVIET UNION                    SWAZILAND                   CAPE VERDE
                        4                            4                            5
    SAO TOME AND PRINCIPE                 SINT MAARTEN                      COMOROS
                        5                            5                            6
                     CUBA                       GAMBIA                         LAOS
                        6                            6                            6
                   MONACO            REPUBLIC OF CONGO                        SAMOA
                        6                            6                            6
> 
```

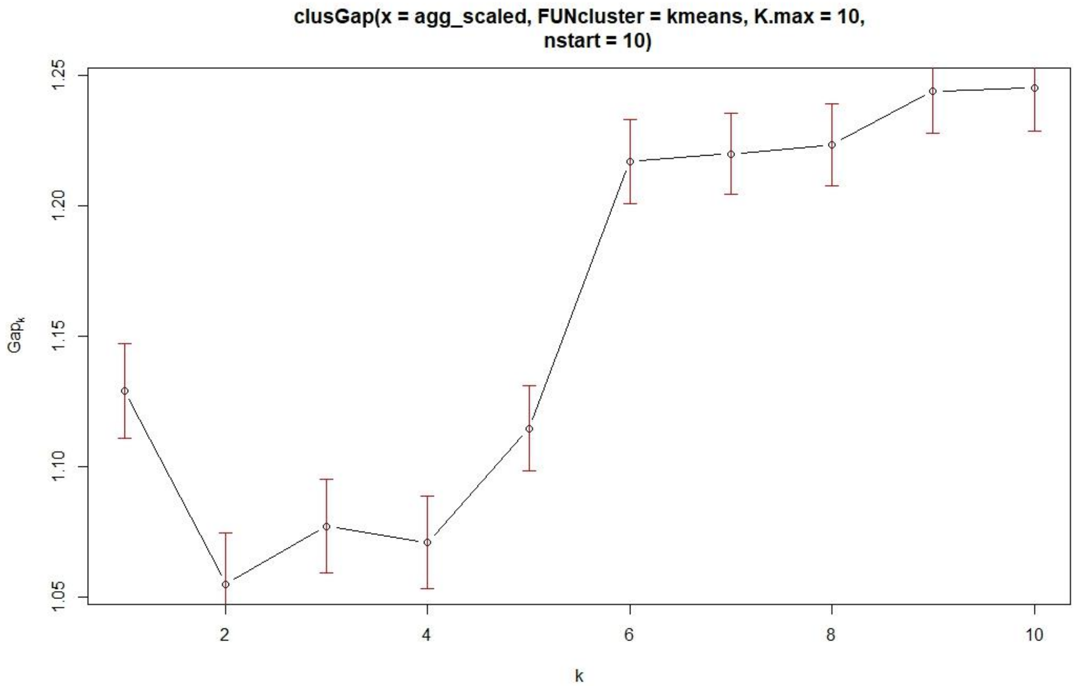
3. Find Optimal Number of Clusters

We first use within sum of squares and look for the knee. And the result shows that 2 is the optimal number of clusters for our hierarchy clustering and 6 is the optimal number of clusters for our k-means.



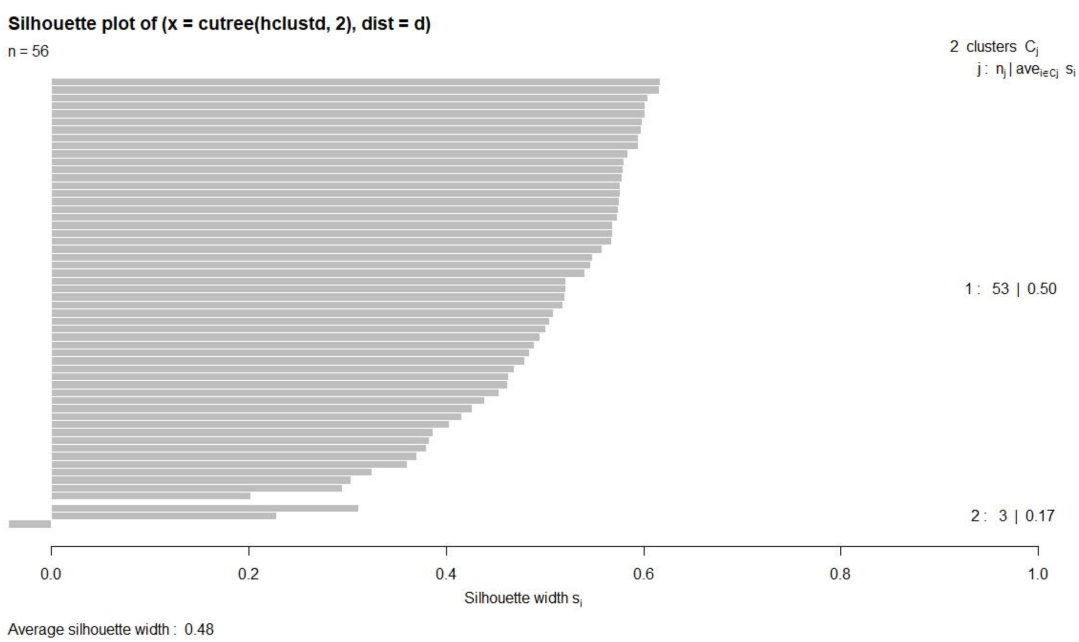


Then we use gap statistic. The result shows that 6 is the optimal number of clusters for our k-means.



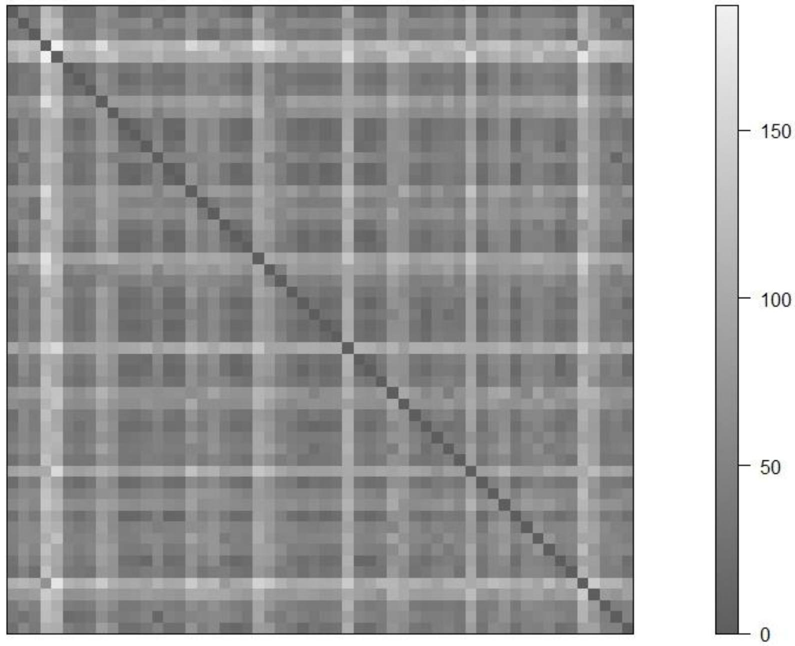
4. Internal Validation

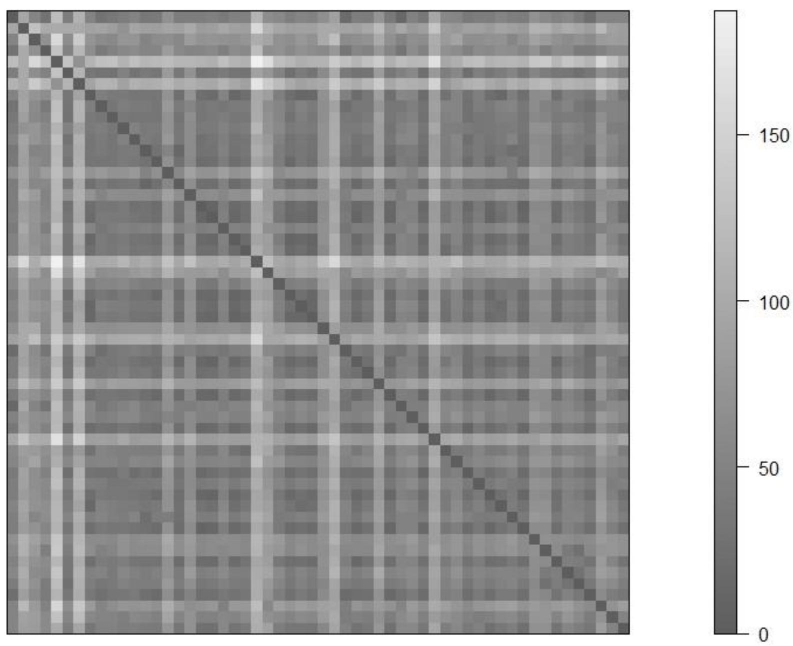
For Class of admission's hierarchy cluster (Silhouette plot)

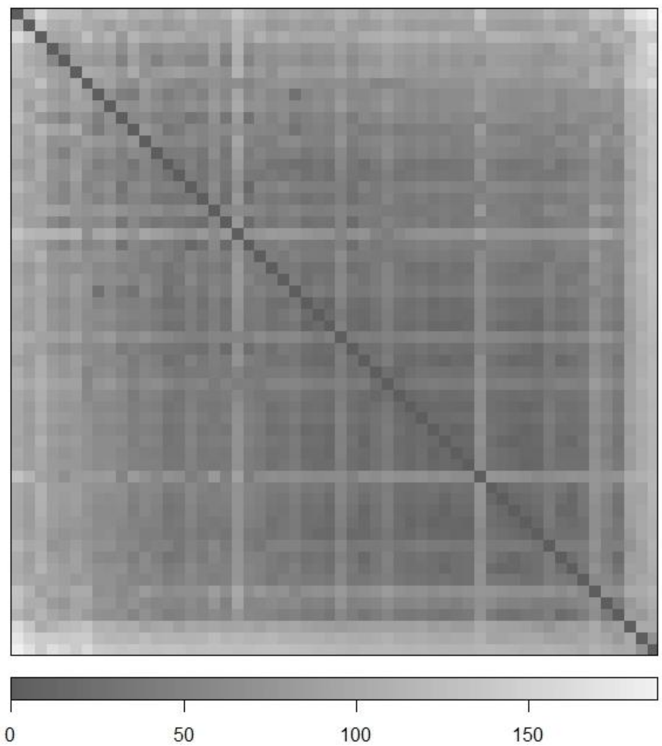


It shows that cluster 1 is the best group while cluster 2 may have some items that are misplaced. The average shows that this may be a good clustering.

For Class of admission's hierarchy cluster (Visualize the Distance Matrix)



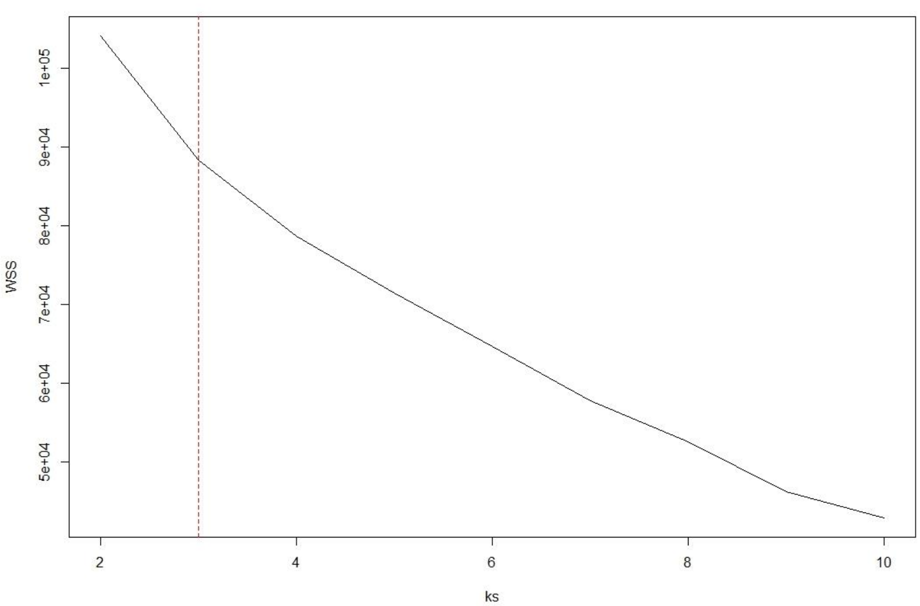




It can be read from the plots' structures that 2 may not be a good value for hierarchy in this condition.

For Class of admission's hierarchy cluster (Compare to other method)

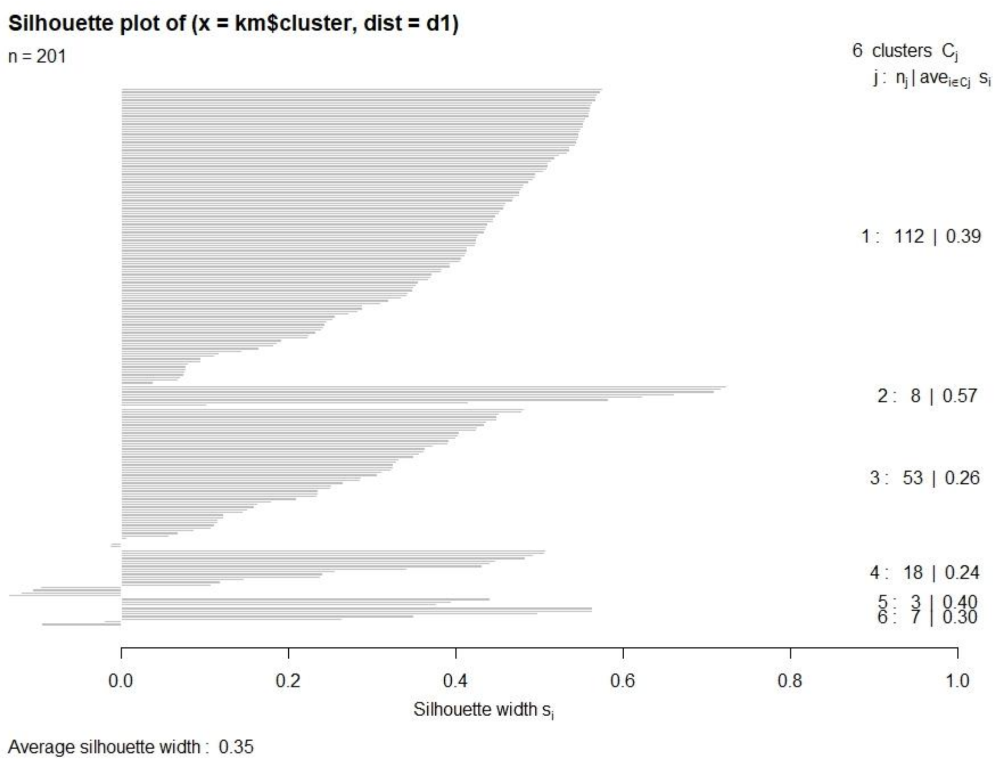
By using within sum of squares, it is easy to find out the optimal k is 3 if using k-means.



```
> sapply(list(
+         kmh=kmh$cluster,
+         h=cluster_h
+         ),
+         FUN=function(x)
+                 fpc::cluster.stats(d, x))[c("within.cluster.ss","avg.silwidth"),]
                   kmh       h
within.cluster.ss 88374.24  104136.2
avg.silwidth      0.2435747 0.4837705
```

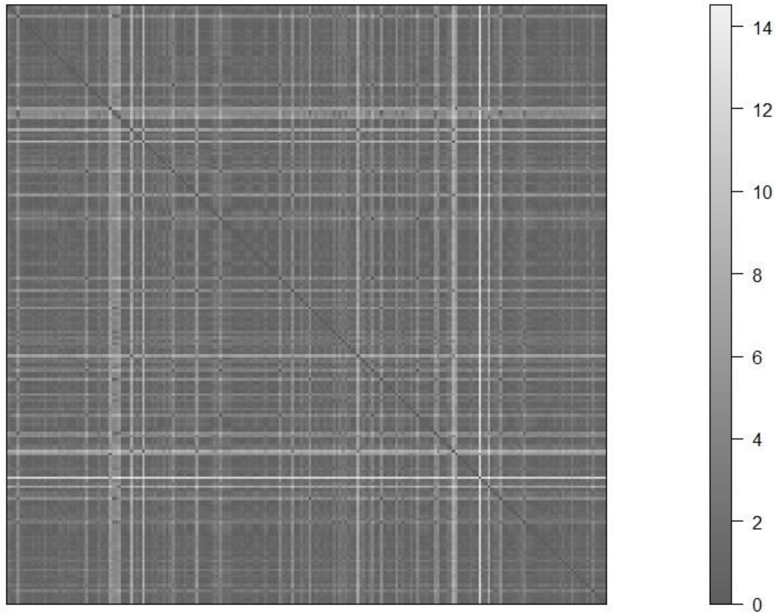
It seems that hierarchy is surely the best choice for this feature.

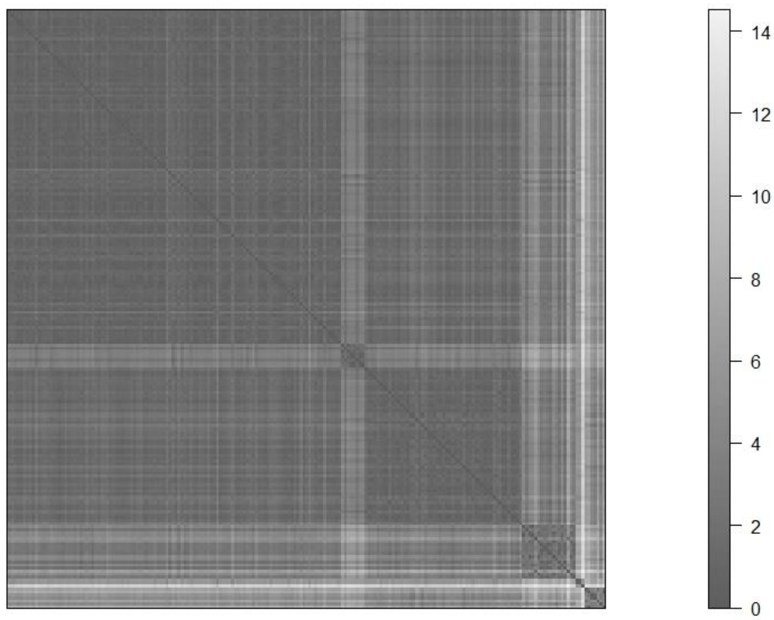
For Country of citizenship's k-means (Silhouette plot)

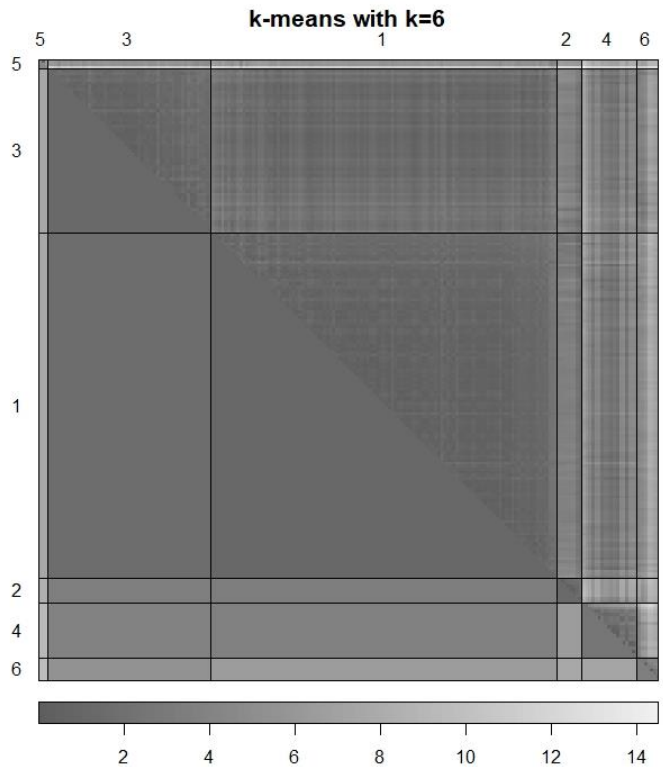


It shows that cluster 2 is the best group while cluster 3, 4, and 6 may have some items that are misplaced. The average shows that this may be a good clustering in some extent.

For Country of citizenship's k-means (Visualize the Distance Matrix)


14
12
10
8
6
4
2
0


14
12
10
8
6
4
2
0

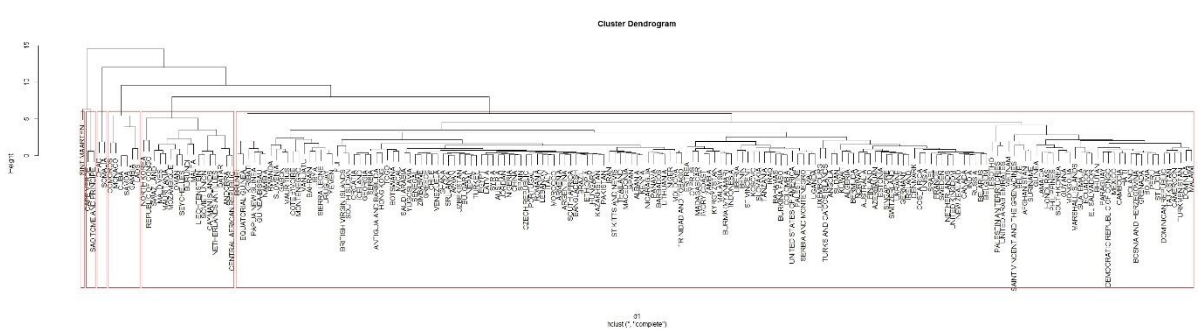


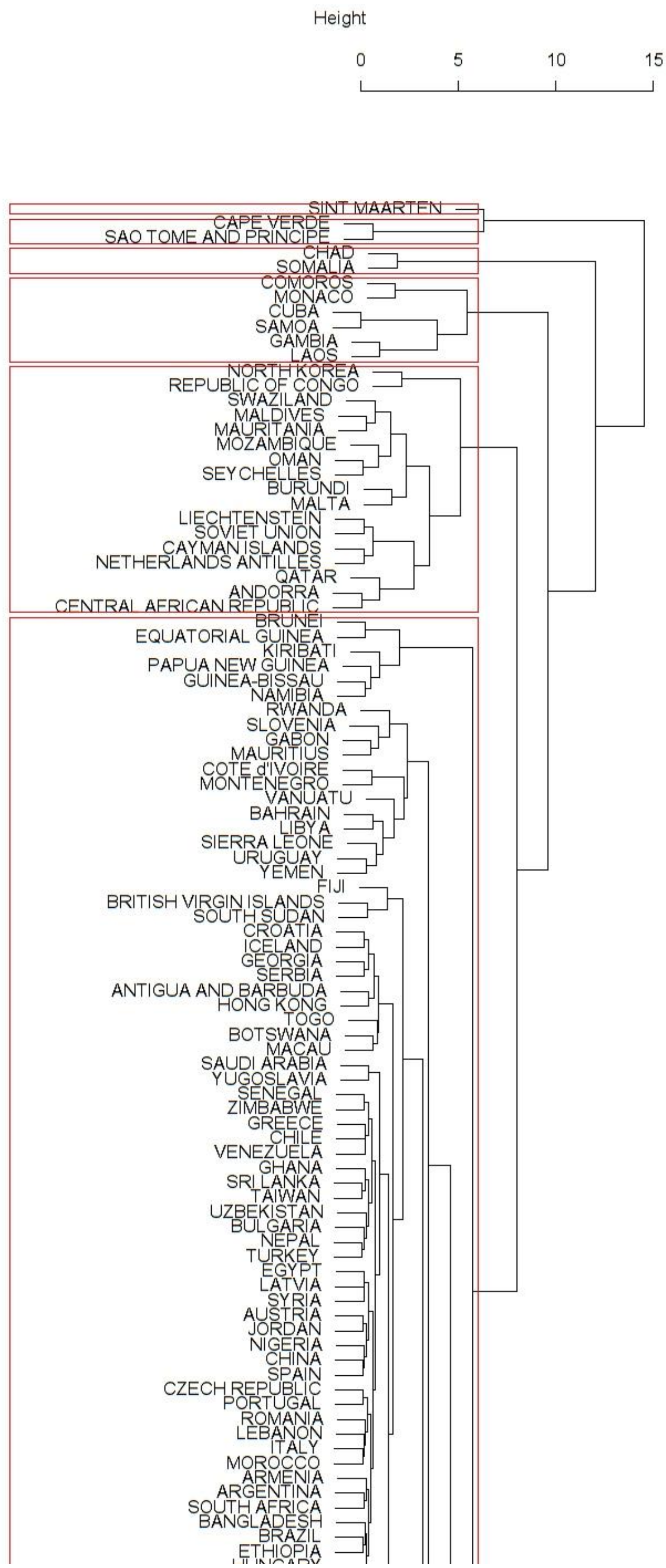
It can be read from the plots' structures that 6 is the best value for k in this condition.

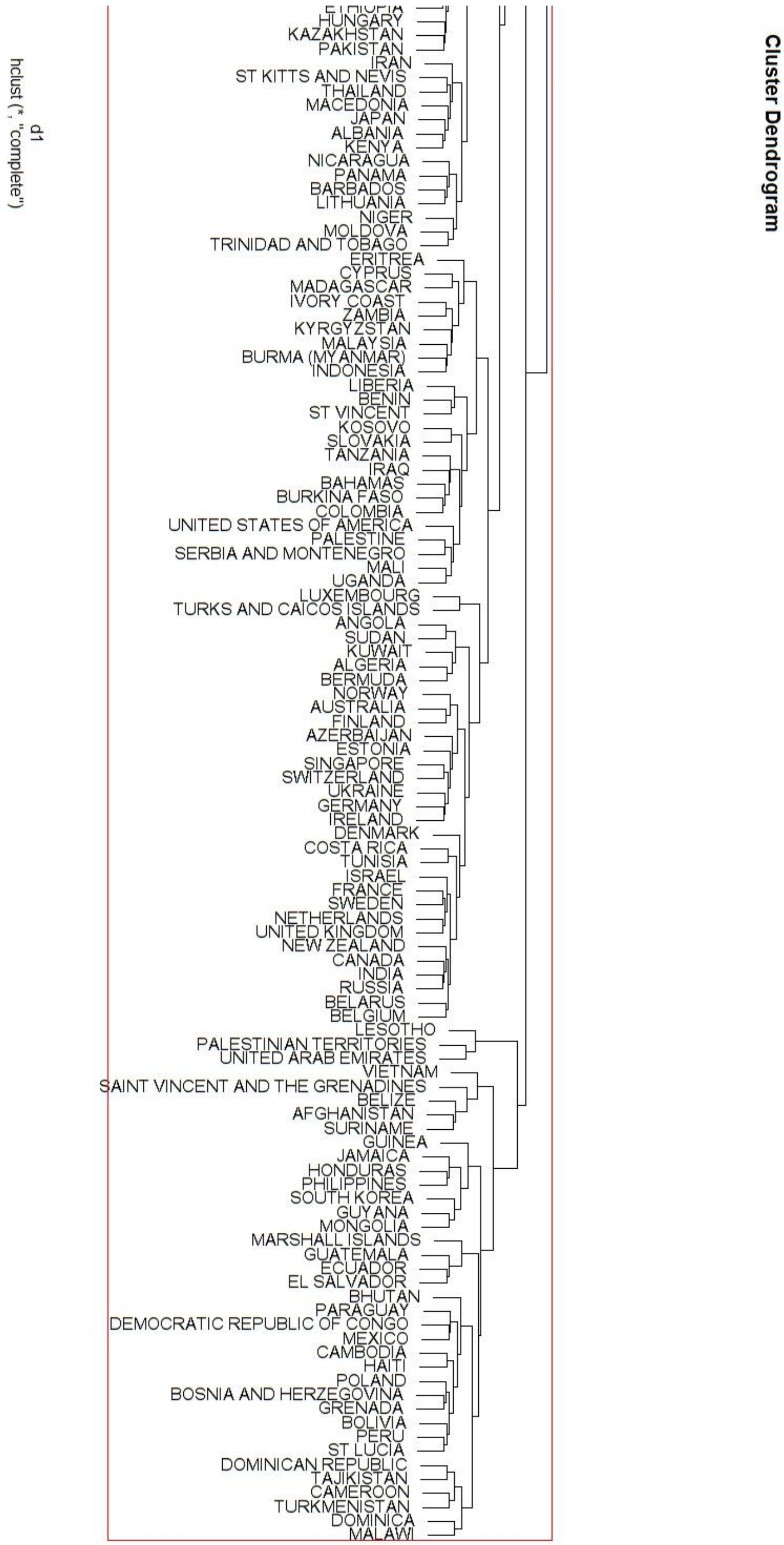
For Country of citizenship's k-means (Compare to other method)

```
> sapply(list(
+     km=km$cluster,
+     hc_compl=cluster_complete
+     ),
+     FUN=function(x)
+         fpc::cluster.stats(d1, x))[c("within.cluster.ss","avg.silwidth"),]
                   km        hc_compl
within.cluster.ss 275.6312  544.0994
avg.silwidth      0.3468048 0.659409
```

It seems that k-means is not the best choice for this feature. So we do clustering again in hierarchy complete. The result is showing as below:




Height
0
5
10
15
SINT MAARTEN
CAPE VERDE
SAO TOME AND PRINCIPE
CHAD
SOMALIA
COMOROS
MONACO
CUBA
SAMOA
GAMBIA
LAOS
NORTH KOREA
REPUBLIC OF CONGO
SWAZILAND
MALDIVES
MAURITANIA
MOZAMBIQUE
OMAN
SEYCHELLES
BURUNDI
MALTA
LIECHTENSTEIN
SOVIET UNION
CAYMAN ISLANDS
NETHERLANDS ANTILLES
QATAR
ANDORRA
CENTRAL AFRICAN REPUBLIC
BRUNEI
EQUATORIAL GUINEA
KIRIBATI
PAPUA NEW GUINEA
GUINEA-BISSAU
NAMIBIA
RWANDA
SLOVENIA
GABON
MAURITIUS
COTE d'IVOIRE
MONTENEGRO
VANUATU
BAHRAIN
LIBYA
SIERRA LEONE
URUGUAY
YEMEN
FIJI
BRITISH VIRGIN ISLANDS
SOUTH SUDAN
CROATIA
ICELAND
GEORGIA
SERBIA
ANTIGUA AND BARBUDA
HONG KONG
TOGO
BOTSWANA
MACAU
SAUDI ARABIA
YUGOSLAVIA
SENEGAL
ZIMBABWE
GREECE
CHILE
VENEZUELA
GHANA
SRI LANKA
TAIWAN
UZBEKISTAN
BULGARIA
NEPAL
TURKEY
EGYPT
LATVIA
SYRIA
AUSTRIA
JORDAN
NIGERIA
CHINA
SPAIN
CZECH REPUBLIC
PORTUGAL
ROMANIA
LEBANON
ITALY
MOROCCO
ARMENIA
ARGENTINA
SOUTH AFRICA
BANGLADESH
BRAZIL
ETHIOPIA


Cluster Dendrogram
d1
hclust (*, "complete")
ETHIOPIA
HUNGARY
KAZAKHSTAN
PAKISTAN
IRAN
ST KITTS AND NEVIS
THAILAND
MACEDONIA
JAPAN
ALBANIA
KENYA
NICARAGUA
PANAMA
BARBADOS
LITHUANIA
NIGER
MOLDOVA
TRINIDAD AND TOBAGO
ERITREA
CYPRUS
MADAGASCAR
IVORY COAST
ZAMBIA
KYRGYZSTAN
MALAYSIA
BURMA (MYANMAR)
INDONESIA
LIBERIA
BENIN
ST VINCENT
KOSOVO
SLOVAKIA
TANZANIA
IRAQ
BAHAMAS
BURKINA FASO
COLOMBIA
UNITED STATES OF AMERICA
PALESTINE
SERBIA AND MONTENEGRO
MALI
UGANDA
LUXEMBOURG
TURKS AND CAICOS ISLANDS
ANGOLA
SUDAN
KUWAIT
ALGERIA
BERMUDA
NORWAY
AUSTRALIA
FINLAND
AZERBAIJAN
ESTONIA
SINGAPORE
SWITZERLAND
UKRAINE
GERMANY
IRELAND
DENMARK
COSTA RICA
TUNISIA
ISRAEL
FRANCE
SWEDEN
NETHERLANDS
UNITED KINGDOM
NEW ZEALAND
CANADA
INDIA
RUSSIA
BELARUS
BELGIUM
LESOTHO
PALESTINIAN TERRITORIES
UNITED ARAB EMIRATES
VIETNAM
SAINT VINCENT AND THE GRENADINES
BELIZE
AFGHANISTAN
SURINAME
GUINEA
JAMAICA
HONDURAS
PHILIPPINES
SOUTH KOREA
GUYANA
MONGOLIA
MARSHALL ISLANDS
GUATEMALA
ECUADOR
EL SALVADOR
BHUTAN
PARAGUAY
DEMOCRATIC REPUBLIC OF CONGO
MEXICO
CAMBODIA
HAITI
POLAND
BOSNIA AND HERZEGOVINA
GRENADA
BOLIVIA
PERU
ST LUCIA
DOMINICAN REPUBLIC
TAJIKISTAN
CAMEROON
TURKMENISTAN
DOMINICA
MALAWI

5. External Validation

We only do the external validation for the country_of_citizenship's k-means.

First, we have to define the ground truth. Cause the ground truth can be simplified as the reality we want our model to predict [1], under this condition, we could define the ground truth as the following target:

The countries be separated into 4 groups, higher wage and higher pass possibility ones get into one group; higher wage and lower pass possibility ones get into one group; lower wage and higher pass possibility ones get into one group; and lower wage and lower pass possibility ones get into one group.

We use the average of the wage and the certified possibility respectively as the standard for grouping. Then the countries can be grouped as this:

```
> length(intersect(cl, wl))
[1] 46
> length(intersect(ch, wl))
[1] 41
> length(intersect(ch, wh))
[1] 85
> length(intersect(cl, wh))
[1] 29
> |
```

(cl: low pass possibility; wl: low wage; ch: high pass possibility; wh: high wage)

Next, we calculate the purity and the entropy. Assume that datasets are grouped into t groups based on our ground truth. C_k is the kth cluster, N is the scale of the data, Ntk is the number of belongings of group t in cluster k, N_k is the size of cluster k. Thus, purity and entropy can be calculated by the following formulate [2]:

$$Pur(C_k) = max \frac{N_{tk}}{N_k}$$

$$Entr\,(C_k) = -\frac{1}{\log(N)} \sum \frac{N_{tk}}{N_k} \log\left(\frac{N_{tk}}{N_k}\right)$$

cluster	clwl---46	chwl---41	chwh--85	clwh--29	entropy	purity
1-112	3	10	83	18	0.156231	0.741071
2----8	0	6	2	0	0.106035	0.75
3-53	27	23	0	2	0.156417	0.509434
4--18	9	0	0	8	0.133311	0.5
5--3	1	2	0	0	0.070417	0.66666
6--7	6	0	0	1	0.077332	0.857143
				mean=	0.116624	0.670718

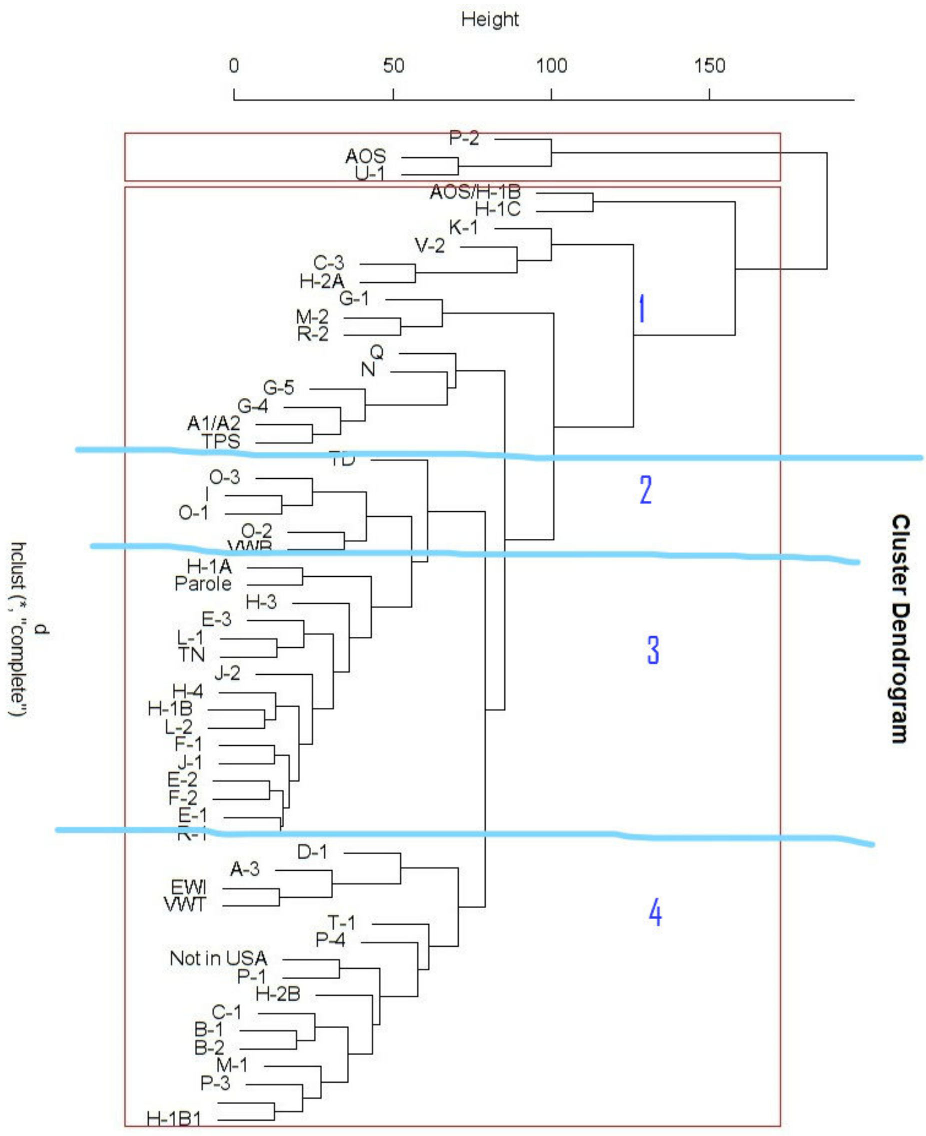
(calculated by MS. Excel)

It is a pretty good result. Entropy is close to 0 while purity is close to 1. That shows the clustering is successed.

V. Evaluation

From the result of the hierarchy clustering of class_of _admission, we could easily get the conclusions like these:

- By judging the case status and the state of the applicants, the Permanent visa applicants could be clustered into 2 groups by the visa type they hold.
- Except the groups of rarely visa holders, the remain can be seen as 4 parts.



1: visa types are related to politics; 2: visa types are related to culture or sports activities;

3: visa types are related to business, academic or working; 4: visa types which are short-term.

From the result of the country_of_citizenship's k-means, we could safely reach the conclusions like follows:

- ❖ By judging the case status and the prevailing wage of the applicants, the residential country of the Permanent visa applicants could be clustered into 6 groups.
- ❖ Beside all groups, the majority is group 1 and 3, which are showing in pink and dark blue points. They share the almost same case status.
- ❖ The other groups can be matches as: 2-black, 4-light blue, 5-green and 6-red.
- ❖ It is obviously that group 1 always own the higher wage than group 3. Combining with the data of economic sector, this may because of the different working types of their majority of emigrations.
- ❖ For group 2, it is a magic group. They are sharing the highest possibility of "certified" and the wage are always near the median! This is because that this kind of countries has rarely green card applicants, their data are more likely to be influenced by any extreme values.
- ❖ For group 4, it is an interesting group, too. They have a uniform distribution of the wage while they have the highest percent of "certified-expired".
- ❖ For group 5, they have a highest percent of "withdraw", and the lower wage. This is the same reason as group 2 (eg.: San Tome And Principe only has two applicants, one got "certified" and another got "withdrawn"), their data are more likely to be influenced by any extreme values.
- ❖ For group 6, most of them share a lowest wage and the highest percentage of "denied". Same to the group 5, their high "denied" is caused by the lowest applicant's number.

VI. Reference

[1] what is the ground truth (https://datascience.stackexchange.com/questions/17839/what-is-ground-truth)

[2] Zhang Weijiao, Liu Chunhuang, Li Fangyu, "Method of Quality Evaluation for Clustering", *Computer Engineering*, Vol.31, Retrieved October, 2005
(https://wenku.baidu.com/view/6d3d9b59804d2b160b4ec01e.html)