# Naloga 1 - Glasovanje za Pesem Evrovizije

Melanija Kraljevska

18 October 2019

## 1 Uvod

Pri tej nalogi bomo analizirali podatke o glasovanju na tekmovanjih za pesem Evrovizije, ki je vsakoletno tekmovanje skladb zabavne glasbe različnih evropskih držav. S pomočjo postopka za hierarhično razvrščanje v skupine, bomo probali ugotoviti osnovne geopolitične in kulturne pristranskosti med državami, ki vplivajo na glasovanje.

## 2 Podatki

Tabela ki vsebuje podatke je na voljo na portalu data.world. V podatkih so vključeni vsa glasovanja iz leta 1975 - 2019 (polfinale in finale), tudi glasovanje žirija in televotinga (ki je od 2016 dalje). Vsebuje 33375 vrstic in naslednje atribute: 'Year', 'Jury or Televoting' , 'From country', 'To country', 'Points'.

### 2.1 Pridobivanje čistih podatkov

Preden podatke spravimo v primerno obliko za obdelavo v nadaljevanju, moramo ustrezno spremeniti podatke, saj nekatere države ne obstajajo več, kot sta Yugoslavia ter Serbia & Montenegro (ju bomo ispustili), nekatere pa imajo različno ime, kot je: F.Y.R Macedonia v 2019 postala North Macedonia. Potem ostanejo 50 različnih držav.

Če pogledamo pravila pri dodelovanju točk: Vsaka država udeleženka dodeli dva sklopa po 12, 10, 8–1 točk svojim 10 najljubšim skladbam: en sklop strokovne žirije in drugi tele glasovanje (če v tistem letu je). Pri tej nalogi bomo obravnavali število točk od 1 do 10, oziroma število točk od 1 do 8 bodo ostali isti, 10 bomo obravnavali kot 9, 12 pa kot 10.

### 2.2 Določanje profile glasovanja

Zdaj ko imamo čiste podatke na voljo, lahko začnemo analizirati glasovanja med državami. Funkcija read_file naj prebere podatke iz tabele in vrne dictionary, ki bo kot key imel ime države, kot value pa vektor velikosti število držav, katere elemente so številke. Vsak element od vektorja, ki bo 'opisal' državo x

vsebuje število točk k, katere je država x dala državi i, v letu j. Če hočemo upoštevati tudi juring/televoting potrebujemo še en parametar. To pomeni da bi tisti vektor imel velikosti: število let * število držav * 2 (juring ali televoting). V tem primeru ne bomo upoštevali žiri in televoting, zarad tega, takrat ko imamo 2 ocene, preposto delimo z 2. Neznanih vrednosti se izogibamo na naslednji način ko bomo računali razdaljo med vektorji, če je vsaj ena koordinata neznana, potem obe ignoriramo. Na koncu povprečimo to razdaljo s številom atributov ki smo jih upoštevali, pa pomnožimo z dolžino vektorja (s tem normaliziramo razdaljo in ohranimo točnost funkcije row_distance pri preverjanju). Še ena zadeva pri kateri moramo bit pozorni je ta, da se lahko zgodi da pri nekatera dva vektorja lahko ignoriramo vse atribute in dobimo razdaljo 0. V tej rešitvi, tisto pojavitev je rešena tako da je temu rezultatu pripejen nek parameter max_razdalja.

# 3 Računanje razdalj

1. row_distance(r1, r2)
   - Za računanje razdalje med dvema vektorja r1 in r2 je uporabljena Euclidska metoda.
   - Razdaljo med vektorja x s koordinatami x1, x2, ..., xn in y s koordinatami y1, y2, ..., yn dobimo tako da najprej seštejemo kvadrate razlike med istoležnimi koordinatami xi in yi, in is te vsote izračunamo koren.
2. cluster_distance(c1, c2)
   - Za računanje razdalje med dvema clustra c1 in c2 je uporabljena metoda average linkage. Torej s pomočjo funkcije row_distance, izračunamo razdalje med vsakega primera iz clustra c1 in vsakega primera iz clustra c2, te razdalje seštejemo, in potem zdelimo s produktom število primerov pri oba clusterja.
   - Glede na to da sta parametri funkcije listi ki so lahko gnezdeni listi, pri tej metodi je tudi uporabljena pomožna metoda (collect_countries()) ki rekurzivno vrne imena držav ki so znotraj posameznega clusterja.
3. closest_clusters(clusters)
   - Ta metoda vrne dva clusterja iz liste clusters pri katerih metoda cluster_distance vrne najmanjšo razdaljo. Poleg tega vrne tudi razdaljo.

# 4 Dendrogram

Za izris tekstovnega dendrograma potrebujemo informacije o tem, kateri clusterji so se med sabo združili. Torej najprej rabimo metodo run() ki bo izvedla

hierarhično združevanje in bo iterativno spremenila listo clusters. Končno listo clusters vsebuje vse podatke o združevanju ki jih potrebujemo da narišemo dendrogram.

1. run()

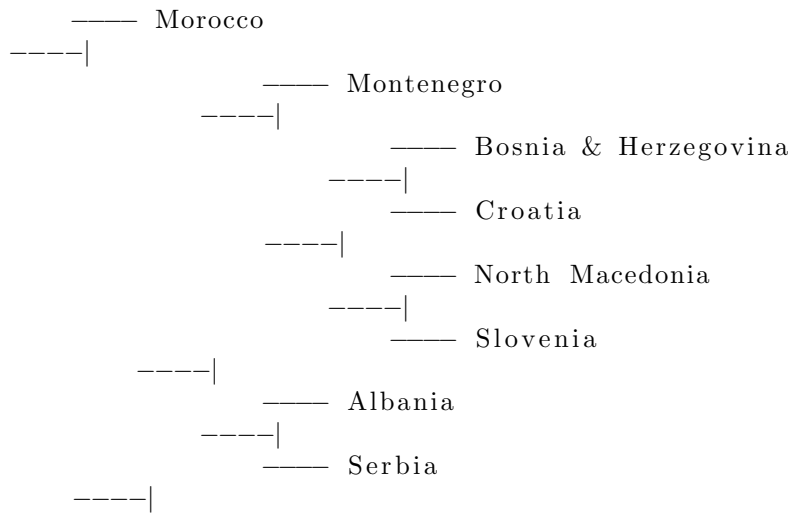
   - Pri tej metodi je uporabljen algoritam združevanja od spodaj navzgor, pri katerem je na začetku vsak primer (država) svoj cluter. Nato izračamo podobnost (oziroma razdaljo) med posameznimi clustri in združimo dva najbolj podobna clusterja, (iz trenutne liste clusterjev odstranimo tista dva ki smo jih združili). To pomeni da bo po vsaki iteracii en cluster manj. Postopek ponavljamo dokler ne ostane samo en cluster.

   - Na koncu izvajanja metode bo lista clusters, ki je dejansko vgnezdena lista iz kje lahko vidimo kako je potekalo združevanje.

2. plot_tree(clusters, level)

   - Rekurzivna metoda ki ima vhodne parametre clusters - katera dva clusterja sta na vrsti, in level - trenutno nivo. Ker združujemo vedno dva clusterja, vemo da vsak cluster ima vedno 2 podclusterja, razen ko ima samo eno državo - takrat se ustavimo.
   - Na začetku je level postavljen na 0, objekt clusters je tisti ki ga je spremenila metoda run(). Torej rekurzivno se sprehajamo po levega in desnega podclusterja in hkrati povečujemo level za 1. Ustavimo se takrat kadar ni več podclusterjev, oziroma pridemo do ime države. Takrat izpišemo ime z določenimi presledki ki jih določa argument level.

   Del dendrograma:




# 5 Skupine in njihove preferenčne izbire

Ko že imamo združene clustre, moramo se odločiti za 'smislen' threshold, ki bo ločeval skupine med seboj. Po par poskusov, je določen threshold 162. Pri tem thresholdu dobimo skupine predstavljene v tabeli.

V final_clusters shranjujemo končne clustre ki so v nadaljevanju predstavljeni v tabeli.

```
threshold = 162
while no_clusters > 1:

        #find the closest clusters and store information
        x, y, d = self.closest_clusters()

        #merge them as one, delete the separate ones
        self.clusters.remove(x)
        self.clusters.remove(y)
        self.clusters.append([x, y])

        #store clusters with distance below threshold
        if d <= threshold:
            self.final_clusters.remove(x)
            self.final_clusters.remove(y)
            self.final_clusters.append([x, y])

        no_clusters -= 1

print('The final clusters are: ', self.final_clusters)
```

Za priljubljene pa nepriljubljene države si lahko pomagamo z izračuna povprečja glasovanja med državami. Pomožna fukncija naj bo imela kot argumente averages - slovar ki za key ima države in kot value ima vektor ki ima kot element povprečno število točk med državo v key in i-ti državi. Ostale argumente v funkciji so: cluster - skupina držav ki jo trenutno obravnavamo, all_countries - lista vseh držav. Oceno za priljubljene oz. nepriljubljene države je 6, pa 2 (v določenih primerih ju lahko prilagodimo da bi dobili smiselne rezultate).

| Skupina | Preferirane države | Nepreferirane države |
|---|---|---|
| Serbia, North Macedonia, Montenegro, Bosnia & Herzegovina, Slovenia, Croatia | Albania, Italy, Turkey, Russia | Armenia, Cyprus, Spain, Finland, Morocco |
| Estonia, Latvia, Lithuania, Finland, Norway, Denmark, United Kingdom, Ireland, Sweden, Poland, Australia | Ukraine, Australia, Bulgaria, Ireland | Albania, Belarus, Morocco, San Marino, Slovakia |
| Romania, San Marino | Greece, Italy | Andorra, Luxembourg, Belarus, Montenegro |
| The Netherlands, Belgium, Austria, Switzerland, Germany | Ireland, Turkey Bulgaria, Bosnia & Herzegovina | Belarus, Montenegro, North Macedonia, Georgia |
| Italy, Morocco | Germany, Malta, Monaco, Turkey, United Kingdom | Armenia, Hungary, San Marino, Slovakia, Slovenia |
| Azerbaijan, Israel, Moldova, Georgia, Belarus, Ukraine, Russia, Armenia, Czech Republic, Bulgaria, Cyprus, Greece | Lithuania, Bulgaria, North Macedonia, Romania | Bosnia & Herzegovina, Luxembourg, Ireland, Albania |
| Portugal, Spain, Andorra | Italy, Bulgaria, Ukraine | Serbia, Poland, Bosnia & Herzegovina |
| Hungary, Monaco | Azerbaijan, France, Israel | Poland, Cyprus, Albania |
| Albania | North Macedonia, Bulgaria, Italy, Greece | Latvia, Iceland, Croatia, Poland |
| France | Italy, Israel, Portugal, Bulgaria | Albania, Montenegro, Morocco, Iceland |
| Luxemburg | Ireland, Malta, United Kingdom | Cyprus, Croatia, Latvia |
| Slovakia | Belgium, Ireland, Malta | Turkey, Czech Republic, Portugal |
| Turkey | Azerbaijan, Italy, Georgia | Poland, Lithuania, Montenegro |
| Malta | Luxembourg, Bulgaria, Malta | Lithuania, San Marino, Germany |

Figure 1: Tabela ki vsebuje skupine držav.

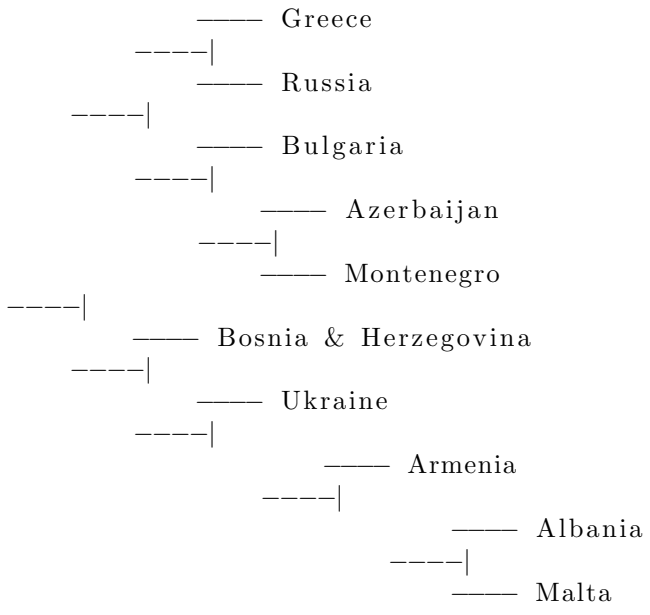
```
def show_statistics(self, cluster, averages, all_countries):

    low_statistics = dict ([])
    high_statistics = dict ([])

    for country in averages:
        low_statistics[country] = 0
        high_statistics[country] = 0

    for country in cluster:
        print("Votes from ", country, ":")
```

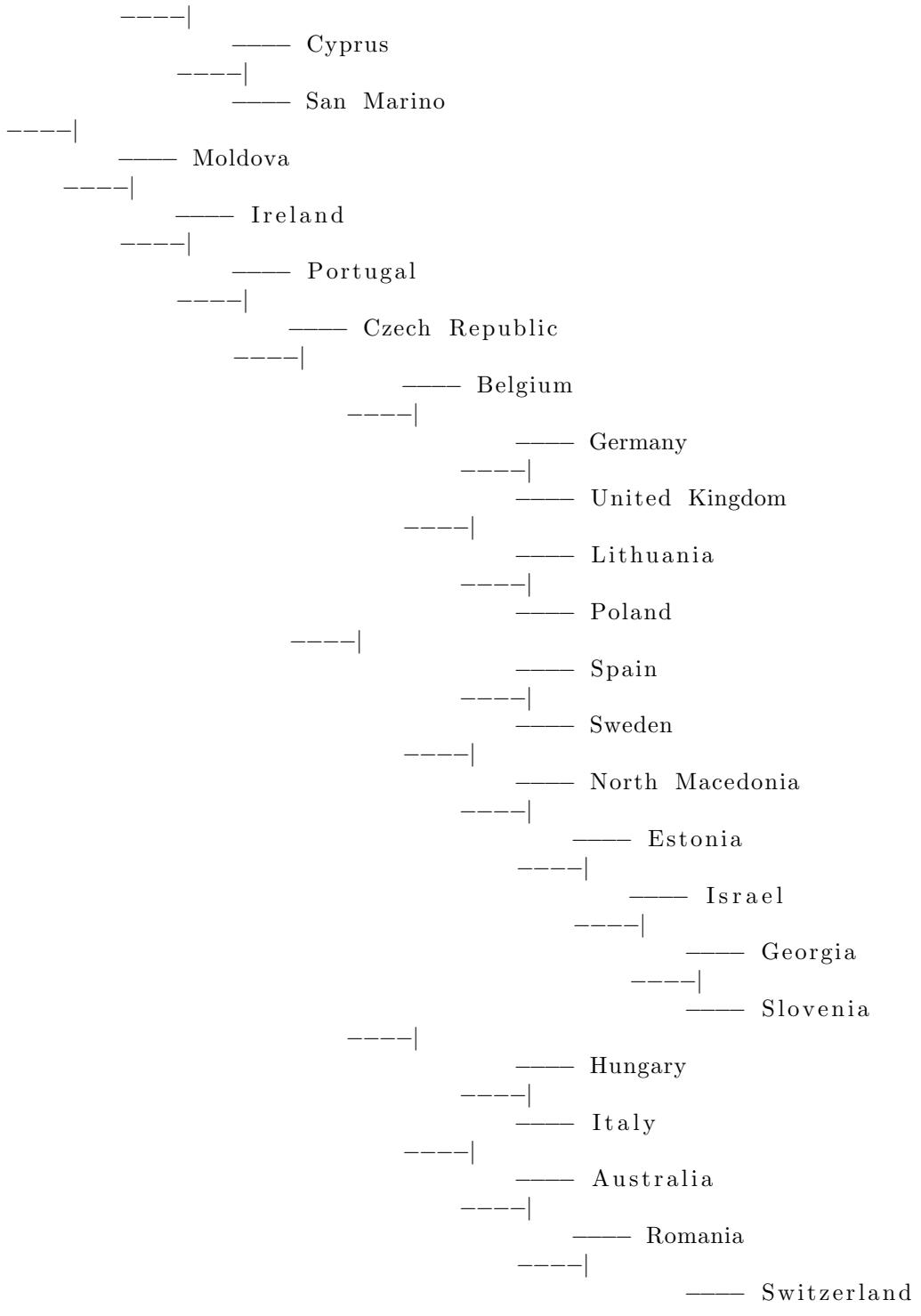
```
            for i in range(len(all_countries)):
                if averages[country][i] >= 6:
                    print("High points: " + all_countries[i])
                    high_statistics[all_countries[i]] += 1
                if averages[country][i] <= 2:
                    print("Low points: " + all_countries[i])
                    low_statistics[all_countries[i]] += 1

    print(high_statistics)
    print(low_statistics)
```

# 6 Žiri in televoting

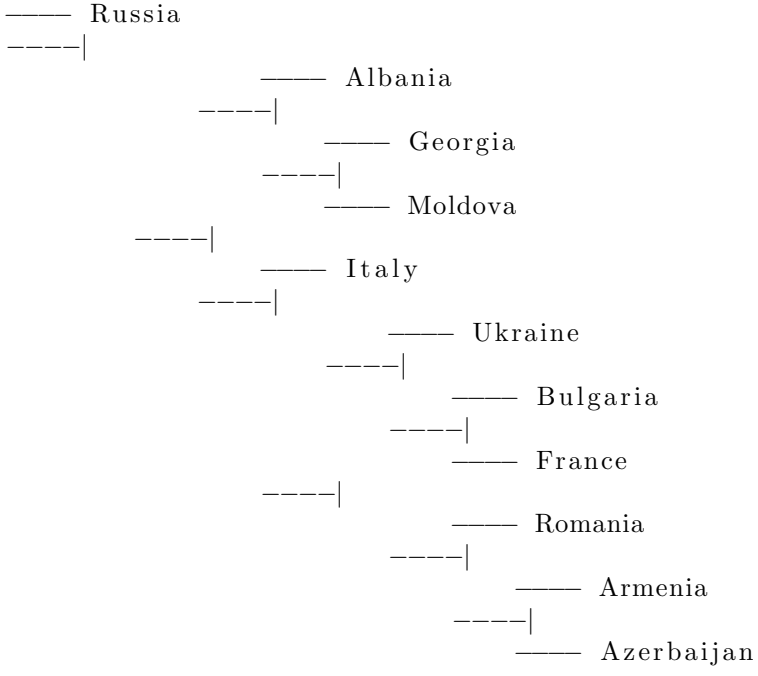
V funkciji read_file_jury_televoting(file_name, parametar) bomo upoštevali samo leta od 2016 dalje. Funkcija naj vrne slovar ki ima za key ime države, kot value pa vektor.Vsak element i iz vektorja države x, predstavlja oceno, kako je država x dala točke državi i. Argument parametar je lahko 'J' ter 'T'. Pri tej funkciji je posamezna ocena določena kot povprečje število točk za leta od 2016 dalje. Ker vsako leto se ne udeležujejo iste države na evroviziji, je pomembno to da upoštevamo kolikokrat se je glasovanje med dvema državama zgodilo. Torej najprej izračunamo skupno število točk ki je država x dala državi i, potem to številko zdelimo s število pojavitev glasovanja (upoštevamo tudi televoing). S tem se izogibamo pojavitve neznanih vrednosti.

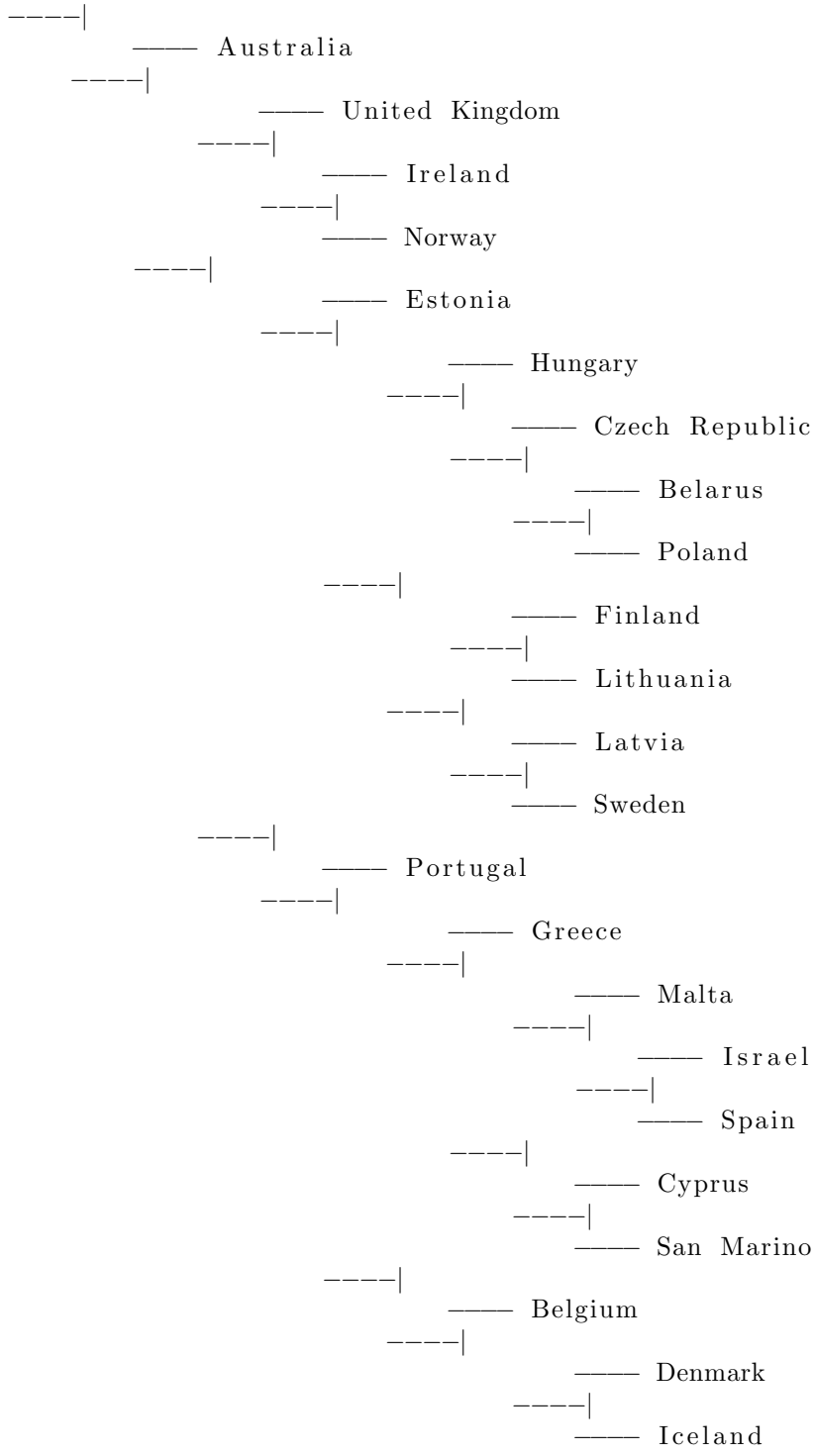
Pogledamo dva dendrograma: Dendrogram glasovanja žirija

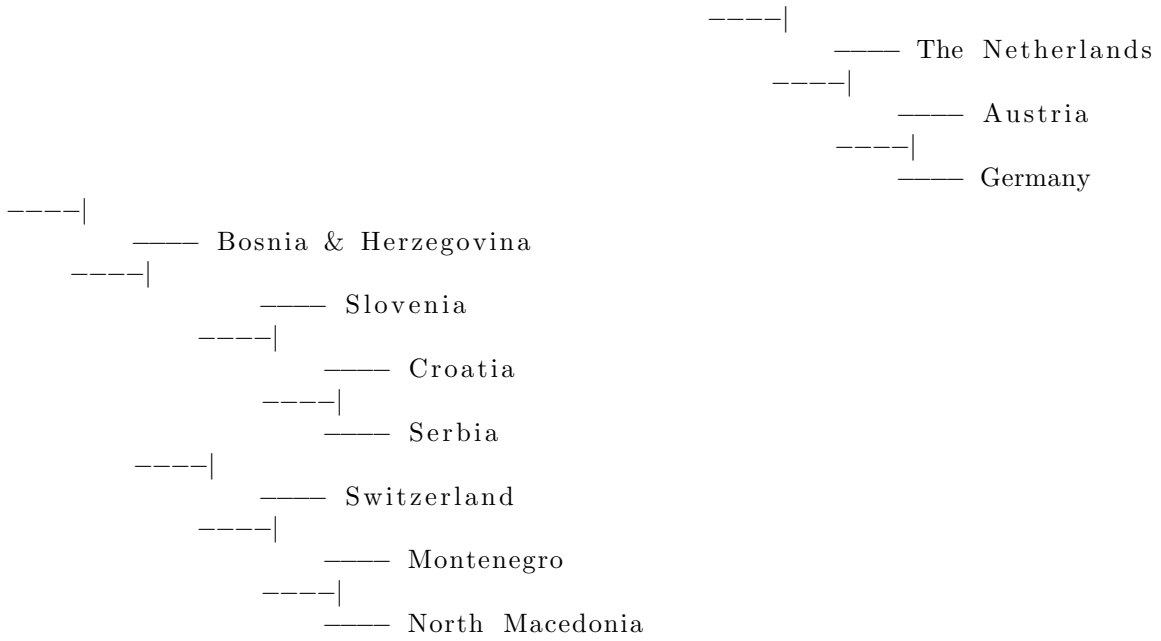



Cyprus
San Marino
Moldova
Ireland
Portugal
Czech Republic
Belgium
Germany
United Kingdom
Lithuania
Poland
Spain
Sweden
North Macedonia
Estonia
Israel
Georgia
Slovenia
Hungary
Italy
Australia
Romania
Switzerland


Dendrogram glasovanja televotinga




Australia
United Kingdom
Ireland
Norway
Estonia
Hungary
Czech Republic
Belarus
Poland
Finland
Lithuania
Latvia
Sweden
Portugal
Greece
Malta
Israel
Spain
Cyprus
San Marino
Belgium
Denmark
Iceland




Lahko opazimo da je glasovanje televotinga bolj pristansko. Tukaj opazimo sosednje države ki so v iste skupine kot so: [Bosnia & Herzegovina, Slovenia, Croatia, Serbia, Montenegro, North Macedonia], [Belgium, Denmark, Iceland, The Netherlands, Austria, Germany], [Latvia, Lithuania],..

Žiri je manj pristanski, vendar tudi se pri njemu najdejo par skupin: [Croatia, Serbia], [Finland, Norway],..