



HACETTEPE
ÜNİVERSİTESİ

Mayıs 2024

GRAFİKSEL VERİ ANALİZİ
FİNAL ÖDEVİ
TREEMAP

Dr. Öğr. Üyesi Onur TOKA

Eda Yaren ÖZEL
2200329007

Ece KANLI
2200329048

1 TREEMAP

1.1 Tarihçe

1990'ların başında Dr. Ben Shneiderman, bilgisayardaki geniş dosya dizinini yeterli şekilde değerlendirmenin bir yolunu bulma zorluğunu çözmeye çalışıyordu. O zamanlar, potansiyel olarak karmaşık klasör yapıları ağı nedeniyle mevcut görselleştirme yöntemleri uygun değildi. Dr. Shneiderman, bu büyük miktarda veriyi (birden fazla düzeyde klasör ve alt klasörle birlikte) ekranda çok fazla yer kaplamadan verimli bir şekilde görselleştirmek için "ağaç haritasını" geliştirdi[1].

1.2 Tanım

Ağaç haritası, herhangi bir hiyerarşik veri kümesine düzenli ve çok düzeyli bir görünüm sunmak için karşılık gelen veri değeriyle orantılı olarak boyutlandırılmış bir dizi iç içe dikdörtgenin kullanıldığı grafikdir [1].

- Hiyerarşi yönü, başlangıçtaki daha büyük dikdörtgenlerin (dallar), daha sonra hiyerarşideki bir sonraki düzeyi temsil eden daha küçük dikdörtgenlerle (alt dallar) döşenmesiyle ortaya çıkar.
- Bir ağaç haritasındaki dikdörtgenler, sol üst köşedeki en büyüğünden başlayarak sağ alttaki en küçüğüne doğru ilerleyerek boyuta göre sıralanır. Birden fazla hiyerarşik düzey görüntülendiğinde bu sıra, iç içe geçmiş dikdörtgenlerin her biri için tekrarlanır.
- Dikdörtgenler bir dizi farklı algoritma kullanılarak boyutlandırılabilir ve sıralanabilir. Her dikdörtgenin boyutunun verinin kendisi tarafından belirlendiğini ve ağaç haritasının sıfır veya negatif değerli veri noktalarını göstermek için kullanılamayacağını unutmamak önemlidir.
- Üst düzey dikdörtgenler arasındaki sınırları vurgulamak için renk kategorik olarak kullanılabilir. Alternatif olarak renk, niceliksel değerleri belirtmek için ısı haritası tarzında uygulanabilir.

Ağaç haritası, iç içe geçmiş dikdörtgenlerden oluşan bir görselleştirme işlevi görür. Bu dikdörtgenler, seçilen bir boyut içindeki belirli kategorileri temsil eder ve bir hiyerarşi veya "ağaç" şeklinde sıralanır. Miktarlar ve desenler sınırlı bir grafik alanında karşılaştırılabilir ve görüntülenebilir. Ağaç haritaları parçadan bütüne ilişkileri temsil eder. Ağaç haritaları, izleyicilerin verilerini bir bakışta yorumlamaları için erişilebilir bir yol sağlar. Renk, boyutları (kategoriler gibi) veya hesaplamaları temsil edebilir. Ağaç haritaları, parça-bütün ilişkisini göstermesi gereken verileri en iyi şekilde gösterir. Her boyut için bir hesaplamanın yüzdeleri, bir araya getirildiğinde bütünü oluşturan kareler halinde görüntülenir[2].

1.3 Kullanım Alanları

- Ağaç haritaları, verilere, mevcut alana ve görselde keşfedilmek veya vurgulanmak istenen ilişkilere bağlı olarak birçok durumda uygun olabilir.
- Bir pasta grafiği üç veya daha az segmentle en iyi şekilde çalışabilirken, bir ağaç haritası birçok segmentle iyi çalışır. Verilerdeki kalıpların görülmesi daha kolaydır, bu da ağaç haritalarının analizin keşif aşamasında kullanılması için iyi bir seçim olmasını sağlar.
- Ağaç haritaları, benzer veya önemli ölçüde farklı görünen kategorilerin ve alt kategorilerin mükemmel bir üst düzey görünümünü sağlar. Rengin verimli kullanımı bu karşılaştırmalara yardımcı olur[1].
- Bir ağaç haritası grafiğindeki dikdörtgenler veya düğümler iki niceliksel değeri göstermemize olanak tanır.
- Bir ağaç haritası grafiğiyle sınırlı bir alanda binlerce veri noktası çizilebilir; diğer veri grafikleri ise alan kısıtlamaları nedeniyle sınırlamalar getirir[3].

1.4 Avantajları/Dezavantajları

Tablo 1: Ağaç Haritasında Gözlenen Durumlar[2]

İyi Ağaç Haritası	Zayıf Ağaç Haritası
<ul style="list-style-type: none">- Farklı sayısal değerler- Farklı bir hiyerarşi- En fazla üç veya dört etiketli karolar- Hiyerarşinin belirgin bir şekilde en üst seviyesi- Negatif değerleri göstermeme	<ul style="list-style-type: none">- Çok fazla kategori ve etiketle dağınık bir görünüm- Çok fazla benzer değerlerin olması

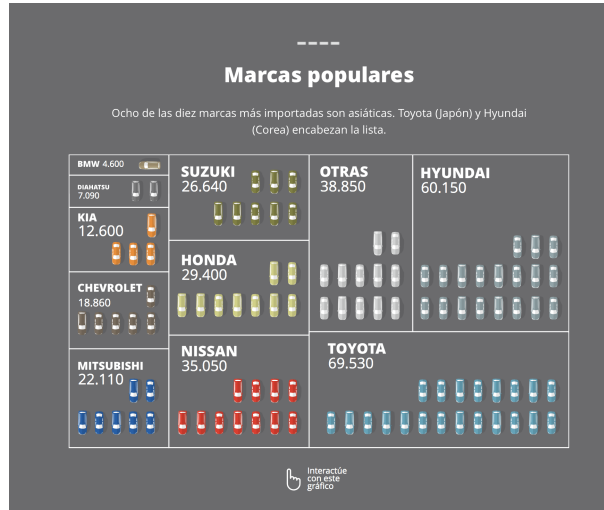
1.4.1 Avantajları

- Ağaç haritaları çok sayıda ilgili kategoriye görselleştirirken faydalıdır[2].
- Ağaç haritası grafiğinin en önemli avantajı, grafikteki dalların ve alt dalların bir bakışta kolayca okunabilmesidir. Veriler ağaç benzeri bir formatta sunulduğunda, büyük miktarda bilgi kompakt ve görsel olarak çekici bir şekilde düzenlenir. Böylece karşılaştırmalar yapmayı ve karmaşık kalıpları parçalamayı kolaylaştırır.
- Ağaç haritası grafikleri, verilerin çeşitli kategorilerini ve alt kategorilerini görüntülemek gerektiğinde son derece faydalıdır. Bir ağaç haritası grafiğini okuyarak, iki değişken arasındaki ilişkinin yanı sıra parça-bütün ilişkileri de kolayca tanımlanabilir. Ağaç haritası grafikleri birkaç katman içerebilir veya verileri onlarca düzeyde temsil edebilir. Böylece yaratıcının yüzlerce alt kategoriye dahil etmesine olanak tanır.
- Binlerce veri noktasının sınırlı bir alanda temsil edilmesi gerektiğinde, ağaç haritası grafikleri mevcut alandan en iyi şekilde yararlanma konusunda etkilidir. Ağaç haritalarını okumayı bu kadar kolay kılan şey, büyük veri kümelerinin en iyi görsel temsili için gerektiği kadar detaya inme ve grafik alanını doldurma yeteneğidir[3].
- Binlerce ögeyi küçük bir ekran alanı içinde görüntüleme yeteneği ağaç haritasının bir avantajıdır. Tek sınırlama, okunaklı bir şekilde etiketlenebilen dikdörtgenlerin sayısıdır[1].

1.4.2 Dezavantajları

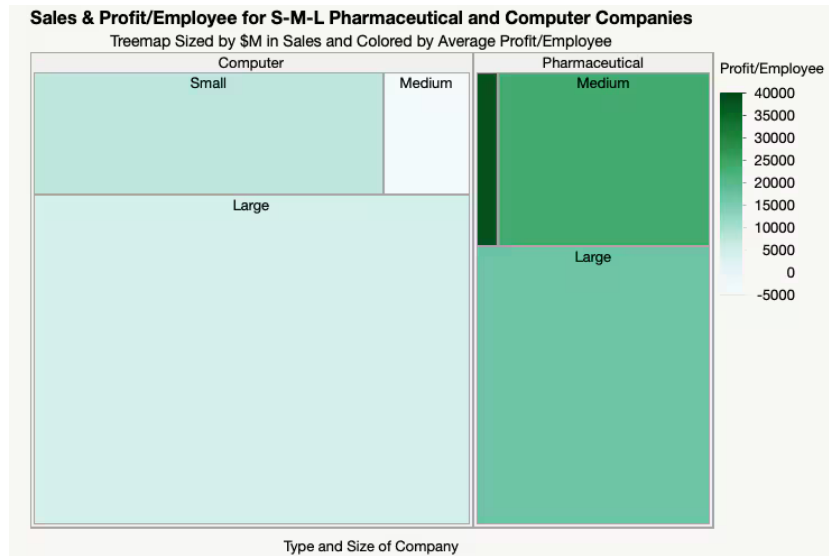
- Ağaç haritaları, doğası gereği, görseldeki daha küçük dikdörtgenlerin etiketlenmesi konusunda zorluk teşkil eder ve bu da çok sayıda isimsiz parçanın ortaya çıkmasına neden olur. Bu nedenle, kullanıcının bu değerleri görüntülemek için bir segmenti tıklayabileceği veya bir segmentin üzerine gelebileceği ve potansiyel olarak daha ayrıntılı bilgi keşfedebileceği, etkileşimli olarak kullanılan ağaç haritalarını görmek daha yaygındır.
- Çoğu durumda, özellikle paylaşmayı seçtiğimiz verileri dikkatli bir şekilde toplayabiliyor veya düzenleyebiliyorsak, başka bir grafik türü verileri daha etkili bir şekilde görüntüleyebilir.
- Veriler önemli bir farklılık göstermiyorsa, ağaç haritamızı benzer ancak aynı olmayan boyutlardaki dikdörtgenlerle doldurursak, bu durum karşılaştırmaları özellikle zorlaştıracaktır.
- Yalnızca her değerın sıfırdan büyük olduğu veri kümeleri için kullanılabilirler. Dikdörtgeni boyutlandırma değeri negatif bir değere sahip olamaz. Bu, potansiyel kullanım durumlarını sınırlar; örneğin kar ve zararı görselleştirilemez[1].

1.5 Örnek Grafikler



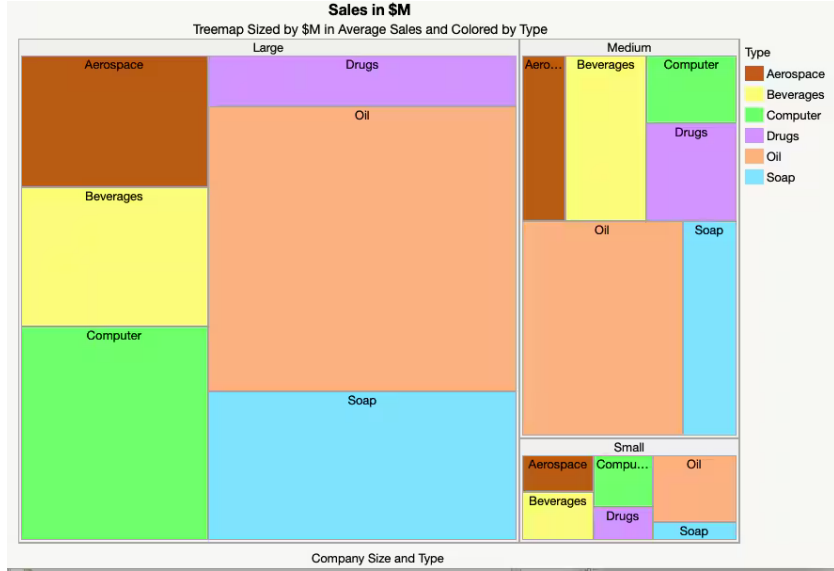
Şekil 1: Popüler araba markaları ve kullanımları[4]

- Şekil 1'deki örnekte popüler araba markalarının verildiği bir treemap grafiği görülmektedir. En çok tercih edilen arabanın Japonya'ya ait olan Toyota ve onu takiben Güney Kore'ye ait olan Hyundai araba markası olduğu görülmektedir.



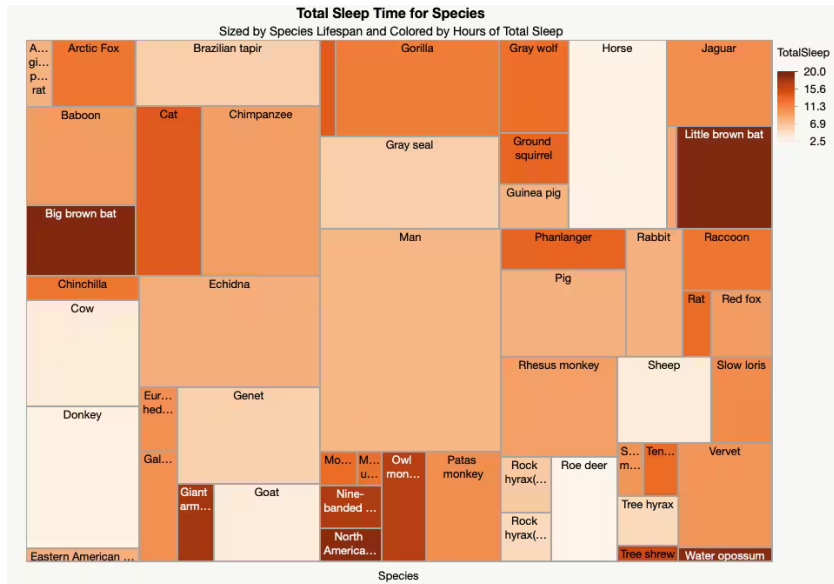
Şekil 2: Şirketler ve satışları[5]

- Şekil 2'deki örnek, kategori ve ağaç haritasına ilişkin bir görseldir. Grafikte, iki şirket kategorisi ve her iki şirkete ait küçük, orta ve büyük firmaların satış yapısı gösterilmektedir. Dikdörtgenler, ABD doları cinsinden ortalama satışlara göre boyutlandırılmış ve kar/çalışan oranına göre renklendirilmiştir.
- Her iki şirkette de büyük firmalar en fazla satışa sahiptir. Computer şirketinde, en yüksek kar/çalışan oranı küçük firmaya aitken orta ölçekli firma da negatif kar/çalışan oranı vardır. Pharmaceutical şirketinde küçük firmaya ait satışların etiketlenmediği görülür. Bu durum, çok sayıda küçük dikdörtgenle sonuçlanan büyük veri kümelerinde yaygındır. En yüksek kar/çalışan oranı orta ölçekli ilaç şirketine ait olduğu görülür.



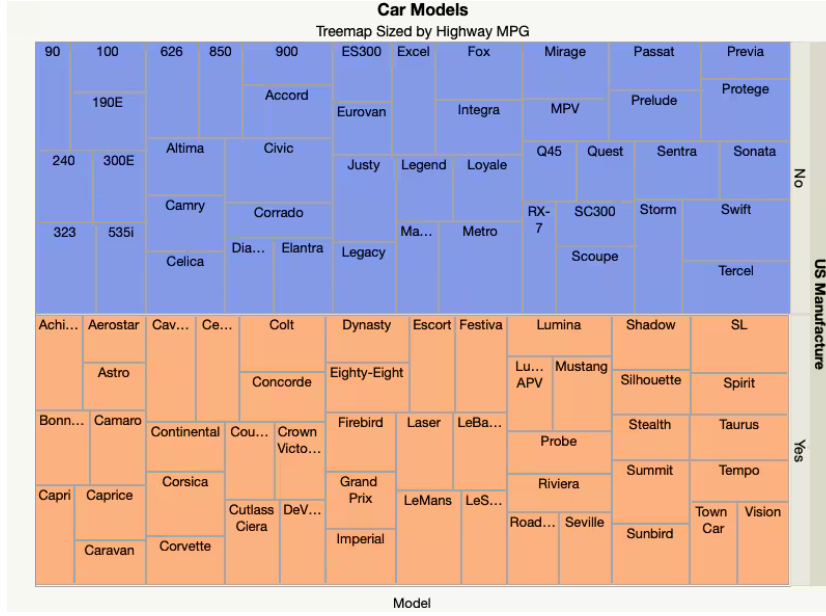
Şekil 3: Farklı seviyelerdeki şirketler ve satışları[5]

- Şekil 3'teki örnek, birçok düzeye sahip daha büyük değişken kümeleri için kategoriler ve hiyerarşiye ilişkin bir görseldir. Grafikte; 6 farklı seviyeye sahip şirketler büyük, küçük ve orta ölçekli şirketler altında verilmiştir. Dikdörtgenler ABD doları cinsinden ortalama satışlara göre boyutlandırılmış ve şirket türüne göre renklendirilmiş ve dikdörtgenler şirket büyüklüğüne göre gruplandırılmıştır.
- Petrol şirketleri, şirket büyüklüğü fark etmeksizin en fazla ortalama satışa sahip şirkettir. Ancak burada ortalama satışlar grafik üzerinde belirtilmediği için en düşük ortalama satışa hangi şirketlerin sahip olduğu net bir şekilde yorumlanamamaktadır.



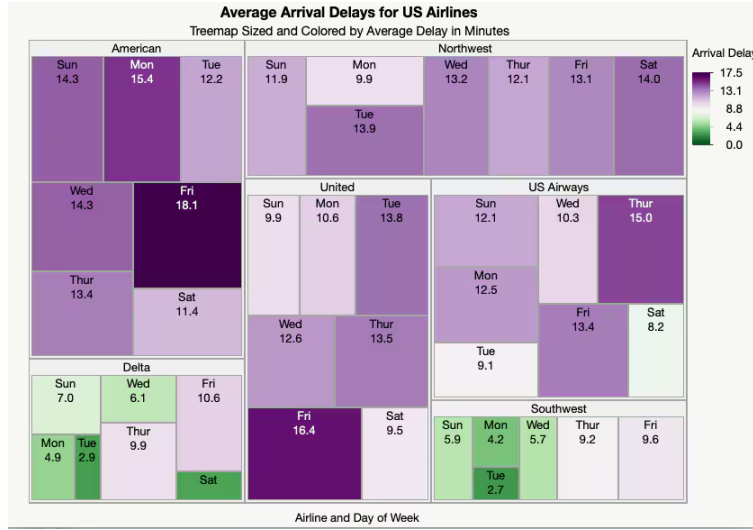
Şekil 4: Canlılar ve uyku süreleri[5]

- Şekil 4'teki örnek, hiyerarşi olmadan çizilen bir ağaç grafiğidir. Grafikte, birçok canlı için saat cinsinden toplam uyku süresini göstermektedir. Dikdörtgenler türlerin yaşam süresine göre boyutlandırılmış ve uyku saatlerine göre renklendirilmiştir.
- Ağaç haritası renklerine göre, yarasalar en uzun toplam uyku süresine sahiptir. Dikdörtgenin boyutuna göre, küçük kahverengi yarasanın büyük kahverengi yarasadan daha uzun bir ömre sahip olduğu söylenebilir. En uzun yaşam süresine insanların sahip olduğu söylenebilir.



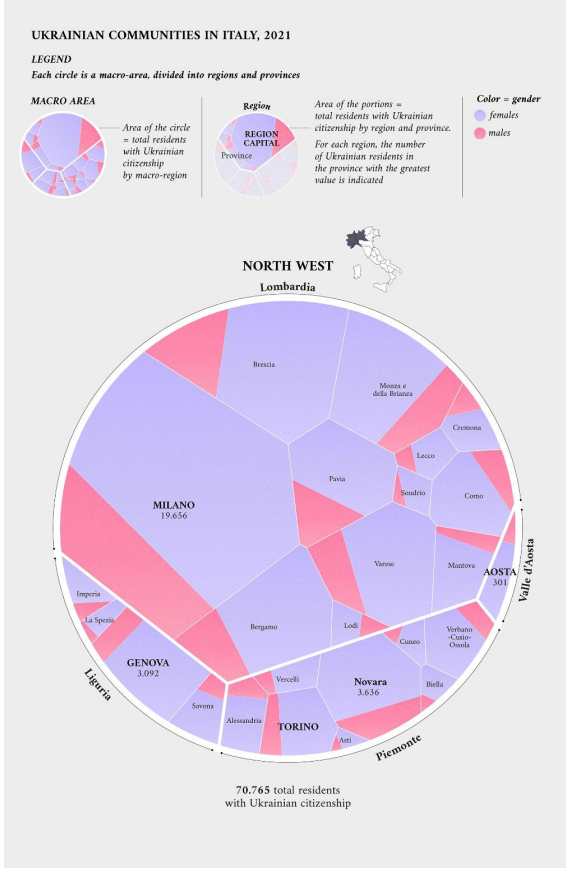
Şekil 5: MPG'ye göre otomobil markaları [5]

- Önceki örnekler kategorileri veya hiyerarşileri x ekseninde gösterirken Şekil 5'deki örnek kategori hiyerarşisini y ekseninde göstermektedir. Veriler 1990'ların ortasındaki otomobillere aittir ve dikdörtgenler otomobil modelleri için galon başına otayol miline (MPG) göre boyutlandırılmıştır. Y eksenini kategori değişkeni, otomobilin ABD'de üretilip üretilmediğini göstermektedir. Ayrıca arabaların alfabetik olarak sıralandığı söylenebilir.
- Metro, tüm arabalar arasında en yüksek MPG'ye sahiptir.



Şekil 6: Havayolu şirketleri ve gecikme süreleri[5]

- Ağaç haritaları, kategorilerin yapısını tanımladığından birden fazla kategori için faydalıdır. Şekil 6'daki örnekte, altı havayolu şirketine ait gecikmeleri ve kategori değişkenleri olarak da haftanın günlerini göstermektedir. Dikdörtgenler ortalama varış gecikmesine göre boyutlandırılmış ve renklendirilmiştir.
- Tüm havayolları için ortalama varış gecikmesi haftanın günlerine göre değişmektedir. Genel olarak en düşük gecikmelere sahip bir havayolları, Southwest veya Delta havayollarıdır. Southwest ve Delta havayollarında ortalama gecikme haftanın başlarında sekiz dakikadan az, Perşembe ve Cuma günleri ise daha fazladır. Bu iki havayolu için en yüksek ortalama gecikme 11 dakikadan azdır. Buna karşılık, American havayolu için en düşük ortalama gecikme 11 dakikadır ve en yüksek ortalama gecikme 18.1 dk ile Cuma gününe aittir.



Şekil 7: Voronoi Afiş[6]

Şekil 7'deki grafiği yorumladığımızda ise, İtalya'nın KuzeyBatı'sında en az Ukrayna vatandaşlığına sahip sakinler, Valle d'Aosta bölgesinde yaşamaktadır. En fazla Ukrayna vatandaşlığına sahip sakinlerin Milano şehrinde ve en az Ukrayna vatandaşlığına sahip sakinlerin Asti şehrinde yaşadığı söylenebilir.

2 VERİ

Ödev kapsamında kullanılan veri seti [7], 1985'den 2016'ya kadar uzanan bir zaman diliminde, ülkelere ait intihar oranlarına ilişkin Tablo 2'deki bilgileri içermektedir.

Tablo 2: Verisetine İlişkin Değişkenler

Değişkenler	
Ülke	Ülke Yılı
Yıl	Yıllık İnsani Gelişim İndeksi
Cinsiyet	Yıllık Gayri Safi Yurtiçi Hasıla
Yaş Aralığı	Gayri Safi Yurtiçi Hasıla Sermaye Yüzdesi
İntihar Sayısı	Nesil
Nüfus	İntihar Oranı

NOT: Oluşturulan grafikler, intihar oranının en yüksek olduğu 1999 yılı referans alınarak oluşturulmuştur.

3 R KODLARININ AÇIKLANMASI

```
# Kütüphanelerin indirilmesi ve çağırılması:
#install.packages("weightedTreemaps")
#install.packages("plotly")
#install.packages("grid")
library(weightedTreemaps)
library(plotly)
library(grid)
library(lubridate)
library(dplyr)
library(forcats)
library(colorspace)
```

Şekil 8: Veri seti için yüklenen R kütüphaneleri

- **install.packages():** Belirtilen paketlerin yüklenmesini sağlar. Bu fonksiyon, CRAN (Comprehensive R Archive Network) gibi bir kaynaktan paketleri indirir ve R ortamına yükler.
- **library():** Yüklenen paketleri R ortamına getirir ve içindeki fonksiyonlara erişimi sağlar.
- **WeightedTreemaps:** Ağırlıklı ağaç haritaları (treemaps) oluşturmak için kullanılan bir kütüphanedir. Bu kütüphane, hiyerarşik verilerin görselleştirilmesi ve veri değerlerine göre alanların ağırlıklandırılması için kullanılır.
- **plotly:** Etkileşimli grafikler ve görselleştirmeler oluşturmak için kullanılan bir kütüphanedir. Bu kütüphane, 2D ve 3D grafikler, etkileşimli bileşenler ve web tabanlı görselleştirmeler sunar.
- **grid:** Izgara tabanlı grafikler oluşturmak ve düzenlemek için kullanılan bir kütüphanedir. Grafiklerin yerleşimini ve düzenini kontrol etmek için güçlü araçlar sağlar ve karmaşık grafik yapıları oluşturmanıza olanak tanır.
- **lubridate:** Tarih ve zaman verileriyle çalışmayı kolaylaştıran bir kütüphanedir. Tarih ve saat analizleri, tarih aritmetiği ve format dönüşümleri gibi işlevler sunar.
- **dplyr:** Veri işleme ve veri manipülasyonu için kullanılan bir pakettir. Veri çerçevelerinde sık kullanılan işlemleri (örneğin, filtreleme, sıralama, toplama) kolaylaştırır ve hızlandırır. Kullanımı kolay bir sözdizimi sağlar ve veri manipülasyonunu daha okunabilir hale getirir.
- **forcats:** Faktör (kategorik) değişkenlerin işlenmesi için kullanılan bir pakettir. Kategorik değişkenlerin düzenlenmesi, yeniden sıralanması ve dönüştürülmesi için kullanışlı işlevler sağlar. Faktör verilerini daha etkili bir şekilde yönetmenizi sağlar.
- **colorspace:** Renk manipülasyonu ve renk paletleri ile çalışmak için kullanılan bir kütüphanedir. Renk dönüşümleri, renk paletleri yaratma ve renk teorisi uygulamaları için işlevler sunar.

```
# Veri setinin yüklenmesi:
suicide_data <- read.csv("master.csv")
head(suicide_data)
```

Şekil 9: Veri setinin yüklenmesi

- **<-:** R programlama dilinde değişken atama operatörü olarak kullanılır. Bu işaret, sağdaki değeri sol taraftaki değişkene atar.
- **read.csv():** CSV (Comma-Separated Values) dosyalarını R'a yüklemek için kullanılır.
- **head():** Belirtilen veri setinin ilk birkaç gözlemini görüntüler. Varsayılan olarak, head fonksiyonu veri setinin ilk 6 satırını gösterir, ancak isteğe bağlı olarak başka bir sayı belirtilebilir.


```

# Veri ön işlemenin yapılması:

#Veri setinde kayıp gözlem kontrolünün yapılması:
colSums(is.na(suicide_data))

#Yıl değişkeninin veri tipinin Date veri tipi yapılması:
attach(suicide_data)
suicide_data$year <- ymd(suicide_data$year, truncated = 2L)

#Kullanılacak değişkenler:country,age,sex:

#Yaş değişkeninin tekil değerlerinin görüntülenmesi:
unique(suicide_data$age)

```

Şekil 10: Veri ön işlemenin yapılması

- **is.na():** Veri setindeki NA (kayıp) değerleri kontrol eder. Her bir hücre için TRUE (NA) veya FALSE (NA değil) değer döner.
- **colSums():** Sütunlar bazında toplam hesaplar. TRUE değerler 1, FALSE değerler ise 0 olarak kabul edilir. Böylece, her bir sütunda kaç adet NA (kayıp) değer olduğunu hesaplar.
- **attach():** Veri setindeki sütun adlarını doğrudan değişkenler olarak kullanılabilir hale getirir. Bu, veri setinin içindeki sütunlara daha kolay erişim sağlar. Ancak, genellikle karmaşadan kaçınmak için dikkatli kullanılması önerilir.
- **\$:** R'da bir veri çerçevesindeki veya listedeki bir sütuna erişmek için kullanılır.
- **ymd():** Lubridate paketinden gelir ve year, month, day formatında tarihleri oluşturur. Ancak burada sadece yıl bilgisi verildiği için, truncated= 2L parametresi kullanılarak yılın yalnızca ilk iki hanesi dikkate alınır ve geri kalan kısımlar varsayılan olarak 01-01 (1 Ocak) olarak kabul edilir. Bu fonksiyon, year değişkenini Date veri tipine dönüştürür.
- **unique():** Belirtilen vektördeki benzersiz (tekil) değerleri döndürür.

```

#Yaş değişkenine karşılık gelen intihar sayıları:
age_intihar <- suicide_data %>%
  group_by(age) %>%
  summarize(toplam_intihar = sum(suicides_no)) %>%
  arrange(desc(toplam_intihar))

#Ülke değişkenine karşılık gelen intihar sayıları:
ulke_intihar <- suicide_data %>%
  group_by(country) %>%
  summarize(toplam_intihar = sum(suicides_no)) %>%
  arrange(desc(toplam_intihar))

#Yıl değişkenine karşılık gelen intihar sayıları:
yil_intihar <- suicide_data %>%
  group_by(year) %>%
  summarize(toplam_intihar = sum(suicides_no)) %>%
  arrange(desc(toplam_intihar))

```

Şekil 11: Veri ön işlemenin yapılması

- **%>%:** Pipe operatörü, bir işlemin çıktısını bir sonraki işleme giriş olarak aktarır. Bu şekilde, kod satırları birbirine bağlanır ve işlem zinciri oluşturulur.
- **group_by():** Veri setini belirtilen değişkenlere göre gruplar. Burada age değişkenine göre gruplama yapılır.
- **summarize():** Gruplanmış veriler üzerinde bir veya daha fazla özet istatistiği hesaplar. Burada toplam intihar sayısı (suicides_no) hesaplanır ve toplam_intihar adı altında özetlenir.
- **sum():** Belirtilen sütundaki tüm değerlerin toplamını hesaplar.

- **arrange():** Veri çerçevesindeki satırları belirli bir sıraya göre düzenlemek için kullanılır. Yani, veriyi belirli bir sütuna veya sütunlara göre artan veya azalan bir şekilde sıralar.
- **desc():** Sıralamanın azalan (büyükten küçüğe) olmasını sağlar. Yani, sıralama işlemi tersine çevrilir.

age	toplam_intihar	year	toplam_intihar	country	toplam_intihar
1 35-54 years	2452141	1 1999-01-01	256119	1 Russian Federation	1209742
2 55-74 years	1658443	2 2002-01-01	256095	2 United States	1034013
3 25-34 years	1123912	3 2003-01-01	256079	3 Japan	806902
4 15-24 years	808542	4 2000-01-01	255832	4 France	329127
5 75+ years	653118	5 2001-01-01	250652	5 Ukraine	319950
6 5-14 years	52264	6 1998-01-01	249591	6 Germany	291262

Şekil 12: Değişkenlerin kategorilerine göre toplam intihar oranları

Kodlar sonucunda, treemap grafiğinde kullanılacak yaş, yıl ve ülke değişkenlerindeki her kategori için toplam intihar sayıları hesaplanıp büyükten küçüğe doğru sıralanarak tablo haline getirilmiştir.

```
# Grafiğin çizdirilmesi:

# Değişkenlerin ayrılması ve toplam intihar oranlarının hesaplanması:
df4 <- suicide_data %>%
  filter(year == "1999-01-01") %>%
  filter(country %in% c("Russian Federation", "United States",
                        "Japan", "France", "Ukraine",
                        "Germany", "Republic of Korea", "Brazil",
                        "Poland", "United Kingdom")) %>%
  filter(age %in% c("35-54 years", "55-74 years", "25-34 years",
                    "15-24 years", "75+ years")) %>%
  filter(sex %in% c("female", "male")) %>%
  # Sadece kadınlar ve erkekler için filtreleme
  group_by(country, sex, age) %>%
  summarise(total_suicides = sum(suicides.100k.pop), .groups = "drop")
```

Şekil 13: Voronoi treemap grafiğinin çizdirilmesi

- **filter():** Belirli bir koşulu sağlayan satırları seçmek için kullanılan bir dplyr fonksiyonudur.
- **== :** Karşılaştırma operatörüdür ve iki değer birbirine eşit olup olmadığını kontrol eder.
- **%in% :** Bir vektörde belirli değerlerin bulunup bulunmadığını kontrol etmek için kullanılan bir operatördür.
- **c():** Bir vektör oluşturmak için kullanılan bir fonksiyondur.
- **.groups:** dplyr fonksiyonlarında kullanılan bir parametredir. Gruplama sonucunun nasıl görüntüleneceğini belirtir. `.groups = "drop"`, gruplama sonuçlarının görüntülenmesini ve adlandırılmasını düzenler.

	country	sex	age	total_suicides
1	Russian Federation	male	55-74 years	99.44
2	Russian Federation	male	35-54 years	98.52
3	Russian Federation	male	75+ years	86.08
4	Ukraine	male	55-74 years	83.74
5	Russian Federation	male	25-34 years	80.83
6	France	male	75+ years	80.46

Şekil 14: Voronoi treemap grafiği için değişkenlerin ayarlanması

Kodların sonucunda, suicide_data veri çerçevesinden yalnızca 1999 yılına ait veriler filtrelenir ve belirli ülkeler, yaş grupları ve cinsiyetler için filtreleme işlemi yapılır. Ardından, bu filtrelenmiş verilere göre her bir ülke, cinsiyet ve yaş grubu kombinasyonu için toplam intihar oranları hesaplanır. Sonuç olarak her bir ülke, cinsiyet ve yaş grubu için toplam intihar oranlarını içeren bir veri çerçevesi oluşturur.

```
# Ülke kategorilerinde cinsiyetlerin ayrılması:
df4 <- df4 %>%
  mutate(sex_within_country = paste(sex, sep = "-"))
# Cinsiyeti ülkelerin içinde ayırarak yeni bir sütun oluşturuyoruz.
```

Şekil 15: Voronoi treemap grafiği için ülke kategorilerine göre cinsiyetlerin ayrılması

- **mutate()**: dplyr paketindeki bir fonksiyondur ve veri çerçevesinde yeni bir sütun oluşturmak veya mevcut sütunları değiştirmek için kullanılır. Veri çerçevesindeki her bir satır için bir işlem yapar ve sonuç olarak veri çerçevesini döndürür.
- **paste()**: Birden fazla karakter dizisini birleştirmek için kullanılan bir fonksiyondur. Özellikle, farklı sütunlardan gelen değerleri birleştirmek için sıkça kullanılır.
- **sep=**: paste() fonksiyonunda kullanılan bir parametredir ve birleştirilen karakter dizilerinin arasına belirtilen ayırıcı bir karakter ekler.

	country	sex	age	total_suicides	sex_within_country
1	Russian Federation	male	55-74 years	99.44	male
2	Russian Federation	male	35-54 years	98.52	male
3	Russian Federation	male	75+ years	86.08	male
4	Ukraine	male	55-74 years	83.74	male
5	Russian Federation	male	25-34 years	80.83	male
6	France	male	75+ years	80.46	male

Şekil 16: Voronoi treemap grafiği için ülke kategorilerine göre cinsiyetlerin ayrılması

Kodların sonucunda, df4 veri çerçevesindeki her bir satır için, ülke kategorilerinde cinsiyetler ayrı bir şekilde birleştirilir. Her bir satırın sex ve country sütunlarındaki değerler, birleştirilerek yeni bir sex_within_country sütununa eklenir.

```
# Her bir cinsiyete birer renk atama:
renkler <- c("female" = "#f9e0fc", "male" = "#d8ebff")

# Renkleri veri çerçevesine ekleme:
df4 <- df4 %>%
  mutate(color = renkler[as.character(sex)])
```

Şekil 17: Voronoi treemap grafiği için her cinsiyet grubuna renk atanması

- **color**: Bir değişken adıdır ve yaş gruplarına atanmış renkleri içerecek bir vektörü ifade eder.
- **as.character()**: Bir nesneyi karakter dizisine dönüştürmek için kullanılır.

```

# Voronoi Treemap oluřturma
tm <- voronoiTreemap(
  data = df4,
  levels = c("age", "country", "sex_within_country"),
  # "country" ve "sex_within_country" sřtunlarını kullanarak
  # hřcreleri břlelim
  cell_size = "total_suicides",
  shape = "circle",
  error_tol = 0.005,
  maxIteration = 200,
  positioning = "clustered_by_area",
  seed = 1
)

```

řekil 18: Voronoi treemap grafiđinin oluřturulması

- **voronoiTreemap():** Belirli bir veri křmesini kullanarak bir Voronoi diyagramı oluřturur ve bu diyagramı bir ađaç yapısı (treemap) řeklinde dřzenler. Bu fonksiyon, verileri belirli seviyelerde gruplayarak, her grubu bir hřcre olarak temsil eder ve hřcrelerin boyutlarını belirli bir metrik ile belirler.
- **data:** Voronoi Treemap'in oluřturulacađı veri çerçevesini belirtir.
- **levels:** Treemap'in hangi deđiřkenlerle oluřturulacađını belirler. Bu parametre, hřcrelerin nasıl břlüneceđini belirler.
- **cell_size:** Her bir hřcrenin boyutunu belirler. Yani, intihar sayılarına gřre hřcre boyutları belirlenir.
- **shape:** Treemap hřcrelerinin řeklini belirler. Burada "circle" deđerı kullanılır, yani hřcreler daire řeklinde oluřturulur.
- **error_tol:** Hřcrelerin birbirine olan toleransını belirler. Křçük bir hata payı belirler.
- **maxIteration:** Voronoi algoritmasının maksimum iterasyon sayısını belirler.
- **positioning:** Hřcrelerin nasıl konumlandırılacađını belirler. "clustered_by_area" deđerı, hřcrelerin alanlarına gřre křmelenerek konumlandırılmasını sađlar.
- **seed:** Rasgele sayı üretiminde kullanılan bir bařlangıç deđeridir. Bu, algoritmanın rastgeleliđini kontrol etmek iin kullanılır.

```

# Treemap izimi
drawTreemap(
  tm,
  color_palette = renkler,
  color_level = 3, # Deđiřken sayısına gřre renklendirme seviyesi
  label_level = c(2), # Sadece řlke isimlerini gřstermek iin
  label_size = 3.3,
  label_color = "black",
  border_color = "white",
  layout = c(1, 1),
  position = c(1, 1)
)

grid.text("1999 Yılında Yař Gruplarına Gřre řlkelerde Kadın-Erkek İntihar Oranları",
  x = 0.5, y = 0.98, gp = gpar(fontsize = 14, fontface = "bold"))

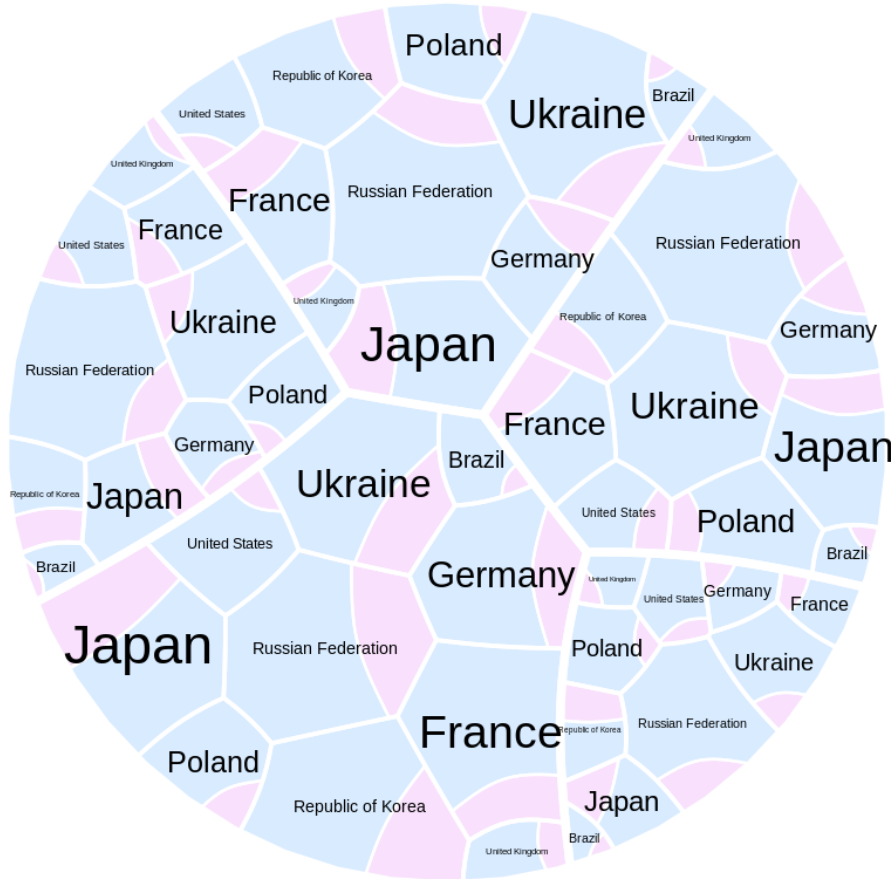
```

řekil 19: Voronoi treemap grafiđinin izdirilmesi

- **drawTreemap():** Voronoi Treemap'in izimini gerekleřtirir.
- **tm:** izilecek Treemap'i belirtir. řnceki adımlarda oluřturulan Treemap nesnesi tm buraya aktarılır.
- **color_palette:** Hřcrelerin renklendirilmesi iin kullanılan renk paletini belirtir. řnceden tanımlanan renkler vektřrř buraya aktarılır.

- **color_level:** Hangi değişkenin renklendirmede kullanılacağını belirtir. Burada 3 değeri kullanılarak cinsiyetin renklendirme seviyesi belirtilir.
- **label_level:** Hangi değişkenin etiketlenileceğini belirtir. Burada sadece ülke isimlerinin (country) gösterilmesi için 2 değeri kullanılır.
- **label_size:** Etiket boyutunu belirtir.
- **label_color:** Etiket renklerini belirtir. Burada siyah renk kullanılır.
- **border_color:** Hücre sınırlarının rengini belirtir. Burada beyaz renk kullanılır.
- **layout:** Çizimin konumunu belirtir. Burada 1,1 değeri kullanılarak tek bir çizim yapılır.
- **position:** Çizimin pozisyonunu belirtir. Burada 1,1 değeri kullanılarak çizimin sol üst köşeye konumlandırılması sağlanır.
- **grid.text():** Çizimin üzerine metin eklemek için kullanılır.
- **x, y:** Bu parametreler, metnin konumunu belirtir. Burada 0.5, 0.98 değerleri kullanılarak metin Treemap'in üst kısmında ve yatay merkezde konumlandırılır.
- **gp:** Metnin görünümünü belirtir. Burada metnin font boyutu ve kalınlığı ayarlanır.

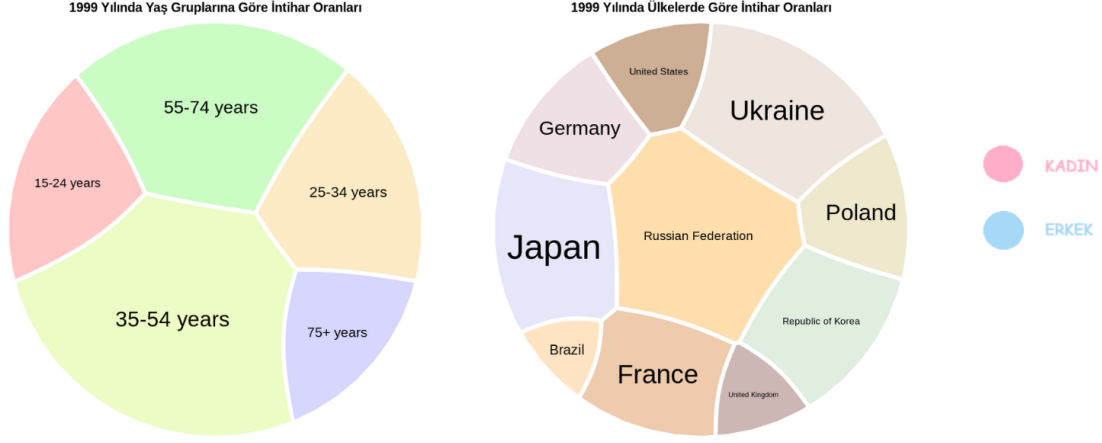
1999 Yılında Yaş Gruplarına Göre Ülkelerde Kadın-Erkek İntihar Oranları



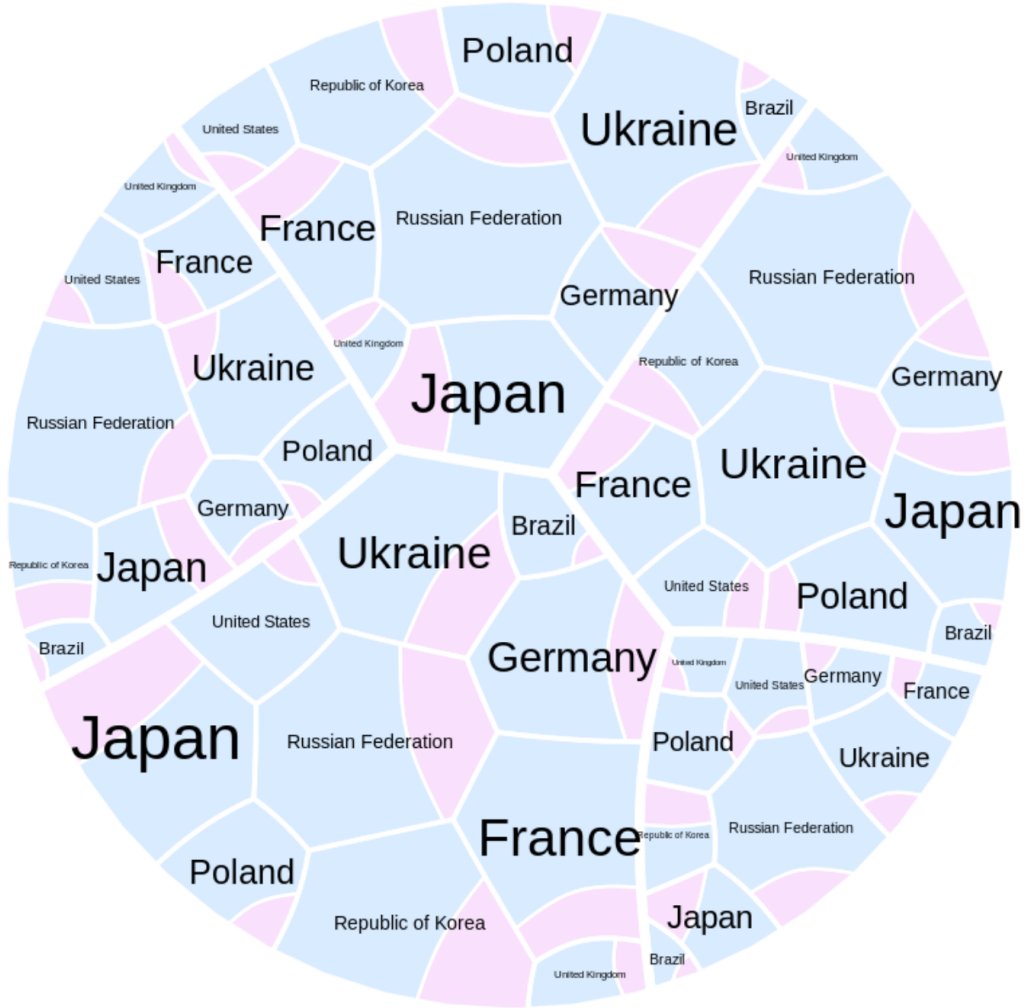
Şekil 20: Sonuç grafiği

4 SONUÇLAR

1999 YILINA AİT İNTİHAR ORANLARININ ÜLKE, YAŞ GRUBU VE CİNSİYET BAKIMINDAN İNCELENMESİ



1999 Yılında Yaş Gruplarına Göre Ülkelerde Kadın-Erkek İntihar Oranları



Şekil 21: Voronoi Treemap Afis

Şekil 21’de, 1999 yılına ait 5 yaş grubu için 10 ülkedeki kadın ve erkek intihar oranlarına dair voronoi ağaç haritası verilmiştir.

Voronoi ağaç haritasını bütünden parçaya doğru yorumladığımızda;

- Voronoi ağaç haritasının tamamı, 1999 yılında 10 ülkede intihar eden kişilerin oranını ifade ediyor.
- Afişin sol üst köşesinde, voronoi ağaç haritasındaki en büyük parçaların yaş gruplarına ait olduğu görülmektedir.
- Onun yanındaki grafikte ise yaş guruplarının alt düzeyleri olan ülkeler verilmiştir. Her ülkenin büyüklüğü intihar oranına göre boyutlandırılmıştır.
- Afişin sağ üst köşesinde yer alan göstergede ise grafikteki renklendirmenin cinsiyete ait olunduğu, pembe rengin kadınları, mavi rengin ise erkekleri simgelediği belirtilmektedir.
- Ana grafik incelendiğinde yaş grupları kendi içinde ülkelere ayrılmış, her ülke ise kendi içinde kadın ve erkek olmak üzere ikiye ayrılmıştır. Etiketlendirilmenin ise ülkelere göre yapıldığı görülmektedir.

Şekil 21’i yorumladığımızda, 1999 yılında en fazla intihar oranının 35-54 yaş aralığında, en az intihar oranının 75+ yaş grubunda olduğu görülmektedir. Her yaş grubu için Rusya Federasyonu en fazla intihar oranına sahip olan ülkedir. Her ülke için erkek intihar oranı, kadın intihar oranından fazla olduğu gözlemlenmektedir.

Kaynaklar

- [1] SIMON ROWE, “what is a treemap?.” <https://www.storytellingwithdata.com/blog/what-is-a-treemap> [Accessed: 13.05.2024].
- [2] No author, “Tree maps.” <https://www.tableau.com/data-insights/reference-library/visual-analytics/charts/treemaps> [Accessed: 13.05.2024].
- [3] No author, “What is a treemapping chart?.” <https://www.jaspersoft.com/articles/what-is-a-treemapping-chart> [Accessed: 13.05.2024].
- [4] No author, “Treemap.” <https://datavizproject.com/data-type/treemap/> [Accessed: 13.05.2024].
- [5] JMP Statistical Software, “Treemap.” https://www.jmp.com/en_us/statistics-knowledge-portal/exploratory-data-analysis/treemap.html [Accessed: 20.05.2024].
- [6] Behance, “Graphics for data europa eu.” https://www.behance.net/gallery/156709667/Graphics-for-Data-EuropaEU?tracking_source=search_projects%7Cfragapane [Accessed: 20.05.2024].
- [7] No author, “Suicide rates overview 1985 to 2016.” <https://www.kaggle.com/datasets/russellyates88/suicide-rates-overview-1985-to-2016> [Accessed: 10.05.2024].