

Mekansal Analiz

Emircan ATALAY

Nisan 2025

Contents

Kütüphaneler	1
Verinin Aktarılması ve Açıklanması	2
Verinin Özet Bilgisi	3
Zamana Göre Deprem Sayısı	3
Türkiye Haritasının Çizdirilmesi	4
Türkiye Haritasının İl Sınırları ile Çizdirilmesi	5
Depremlerin Harita Üzerindeki Gösterimleri	6
İnteraktif Noktasal Harita	6
Quadrat Analizi	6
Hotspot Grafiği	8
Yoğunluk Haritası(KDE)	9
Silhouette Yöntemi	11
Komşuluk Testleri	13
Moran I Testi	13
Ripley K	14
LISA Testi	15
K-Distance Testi	16
DBSCAN Kümeleme Analizi	17
Büyükölçeklerine Göre Hotspot Grafiği	19
Kaynakça	20

Kütüphaneler

```
#install.packages(c("ggplot2", "sf", "tmap", "leaflet", "dplyr", "rnaturalearth", "rnaturalearthdata", "leaflet",  
# "lubridate", "geodata")  
library(tmap)  
library(ggplot2)
```

```
library(sf)
library(leaflet)
library(readr)
library(dplyr)
library(readxl)
library(rnaturalearth)
library(rnaturalearthdata)
library(leaflet.extras)
library(spdep)
library(sp)
library(dbscan)
library(lubridate)
library(geodata)
library(geosphere)
library(cluster)
library(FNN)
library(car)
library(spatstat)
```

Verinin Aktarılması ve Açıklanması

```
library(readxl)
a = read_excel("C:/Users/emirc/Desktop/MEKANSAL/orijinalveri.xlsx")
head(a)
```

```
## # A tibble: 6 x 15
##       No `Deprem Kodu` `Olus tarihi`      `Olus zamani` Enlem Boylam `Der(km)`
##   <dbl>      <dbl> <dtm>          <chr>          <dbl> <dbl>    <dbl>
## 1     1        2.02e13 2024-06-30 00:00:00 18:55:19.36   38.1   38.4      5
## 2     2        2.02e13 2024-06-30 00:00:00 18:02:35.51   39.3   27.5     9.6
## 3     3        2.02e13 2024-06-30 00:00:00 15:53:24.98   37.8   27.0     5.4
## 4     4        2.02e13 2024-06-30 00:00:00 12:50:55.42   38.0   27.1     7.4
## 5     5        2.02e13 2024-06-30 00:00:00 10:53:54.18   40.2   31.8      5
## 6     6        2.02e13 2024-06-30 00:00:00 09:54:20.15   38.6   43.0    24.4
## # i 8 more variables: xM <dbl>, MD <dbl>, ML <dbl>, Mw <dbl>, Ms <dbl>,
## #   Mb <dbl>, Tip <chr>, Yer <chr>
```

Veri seti, 2024 yılı Mayıs ve Haziran aylarında Türkiye’de meydana gelen, büyüklüğü 2 ve üzerindeki depremleri kapsamaktadır. Veriler, Türkiye’nin çeşitli bölgelerinde meydana gelmiş 1315 adet gözlemden oluşmaktadır.

Gerekli görülen değişken isimleri aşağıdaki gibi güncellenmiştir.

```
a_sf = st_as_sf(a, coords = c("Boylam", "Enlem"), crs = 4326)
names(a_sf)[names(a_sf) == "Olus tarihi"] = "Tarih"
names(a)[names(a) == "Olus tarihi"] = "Tarih"
names(a)[names(a) == "Olus zamani"] = "Zaman"
```

Verinin Özet Bilgisi

```
summary(a[, c(5,6,7, 10)])
```

```
##           Enlem           Boylam           Der(km)           ML
##  Min.      :35.03   Min.      :26.00   Min.      : 0.000   Min.      :2.000
## 1st Qu.:37.75   1st Qu.:28.89   1st Qu.: 5.000   1st Qu.:2.100
## Median :38.31   Median :36.42   Median : 6.400   Median :2.300
## Mean   :38.35   Mean   :34.56   Mean   : 8.836   Mean   :2.422
## 3rd Qu.:39.18   3rd Qu.:38.21   3rd Qu.: 9.900   3rd Qu.:2.600
## Max.   :41.88   Max.   :44.66   Max.   :146.200   Max.   :4.900
```

Veri setindeki ML değerleri incelendiğinde, deprem büyüklüğünün en küçük değeri 2.0, en büyük değeri ise 4.9 olarak gözlemlenmiştir. Depremlerin derinlik (km) değerlerine bakıldığında ise en derin deprem 146.2 km olarak ölçülmüştür.

Zamana Göre Deprem Sayısı

Veri setindeki toplam 1315 deprem, günlük bazda filtrelenerek analiz edilmiştir. Günlük deprem sıklığına bakıldığında, her bir gün içinde meydana gelen deprem sayısının zamanla nasıl değiştiği aşağıda incelenmiştir.

```
a$Tarih = as.Date(a$Tarih, format = "%d.%m.%Y")
```

```
gunluk_deprem_sayisi = a %>%
  group_by(Tarih) %>%
  summarise(deprem_sayisi = n())

head(gunluk_deprem_sayisi)
```

```
## # A tibble: 6 x 2
##   Tarih      deprem_sayisi
##   <date>          <int>
## 1 2024-05-01             21
## 2 2024-05-02             28
## 3 2024-05-03             22
## 4 2024-05-04             19
## 5 2024-05-05             23
## 6 2024-05-06             9
```

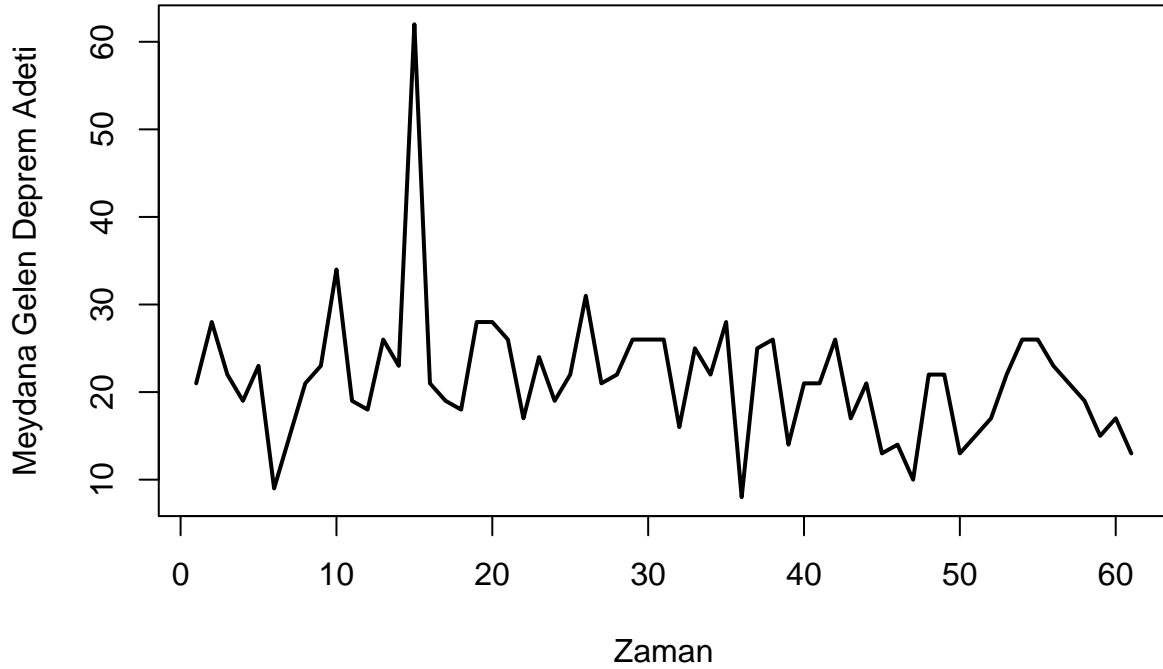
```
summary(gunluk_deprem_sayisi)
```

```
##           Tarih           deprem_sayisi
##  Min.      :2024-05-01   Min.      : 8.00
## 1st Qu.:2024-05-16   1st Qu.:17.00
## Median :2024-05-31   Median :21.00
## Mean   :2024-05-31   Mean   :21.56
## 3rd Qu.:2024-06-15   3rd Qu.:26.00
## Max.   :2024-06-30   Max.   :62.00
```

Bu tarihler arasında Türkiye ve yakın çevresinde meydana gelen depremlerin günlük sayıları incelendiğinde, bir gün içerisinde en az 8 deprem, en fazla ise 62 deprem kaydedilmiştir. Günlük ortalama deprem sayısı ise 21 olarak hesaplanmıştır.

```
veri_ts = ts(gunluk_deprem_sayisi$deprem_sayisi, frequency = 1)

ts.plot(veri_ts, gpars=list(xlab="Zaman", ylab="Meydana Gelen Deprem Adeti", lwd=2))
```



61 günlük süre boyunca, günlük toplam deprem sayıları ortalama olarak birbirine yakın değerler göstermektedir. Ancak, 15. günde meydana gelen 62 deprem, diğer günlere göre belirgin bir artış göstermektedir.

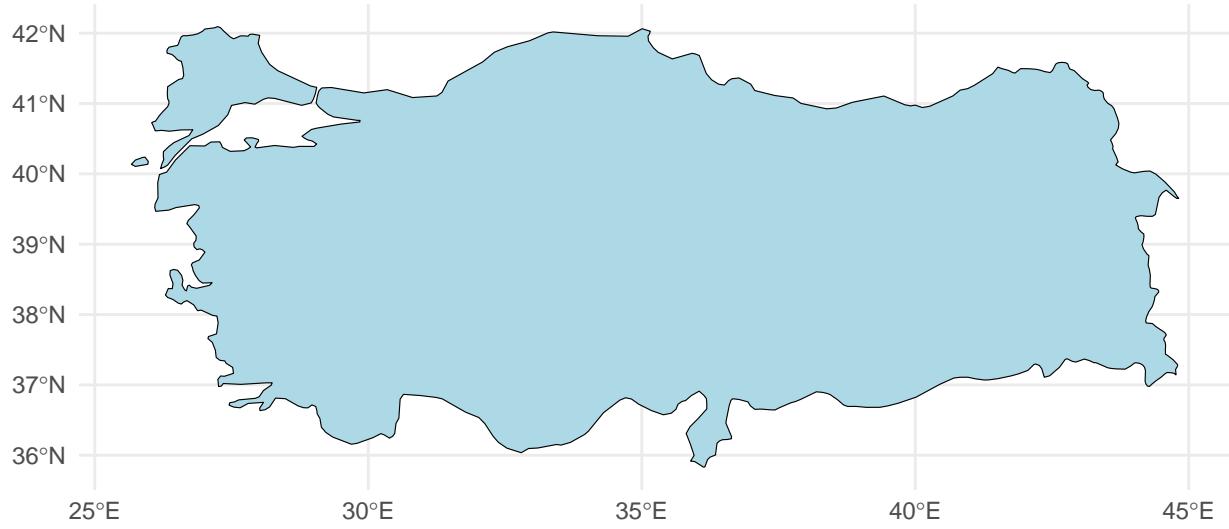
Türkiye Haritasının Çizdirilmesi

```
world = ne_countries(scale = "medium", returnclass = "sf")

turkey = world[world$name == "Turkey", ]

ggplot(data = turkey) +
  geom_sf(fill = "lightblue", color = "black") +
  ggtitle("Türkiye Siyasi Sınırları") +
  theme_minimal()
```

Türkiye Siyasi Sinirlari

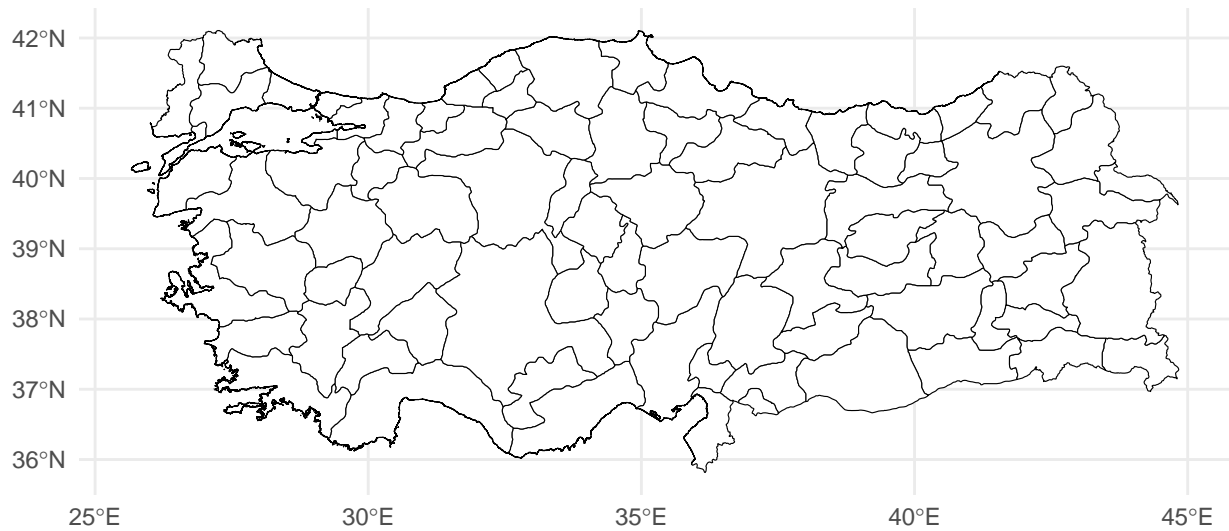


```
turkey_map = map_data("world", region = "Turkey")
```

Türkiye Haritasının İl Sınırları ile Çizdirilmesi

```
turkiye_il = geodata::gadm(country = "TUR", level = 1, path = tempdir())  
  
turkiye_il_sf = st_as_sf(turkiye_il)  
  
ggplot(data = turkiye_il_sf) +  
  geom_sf(fill = "white", color = "black") +  
  ggtitle("Türkiye İl Sınırları") +  
  theme_minimal()
```

Türkiye İl Sinirlari



Depremlerin Harita Üzerindeki Gösterimleri

İnteraktif Noktasal Harita

```
#leaflet(a_sf) %>%
# addTiles() %>%
#addCircleMarkers(
#radius = ~ML * 1.5,
#color = ~ifelse(ML >= 3.5, "red", "orange"),
#popup = ~paste("Yer:", Yer, "<br>buyukluk:", ML, "<br>Tarih:", Tarih)
# )
```

Yukarıdaki haritada, veri setindeki tüm depremler büyüklüklerine ve konumlarına göre görselleştirilmiştir. Kırmızı renk ile gösterilmiş olan noktalar, $ML > 3.5$ büyüklüğündeki depremleri temsil etmektedir.

Quadrat Analizi

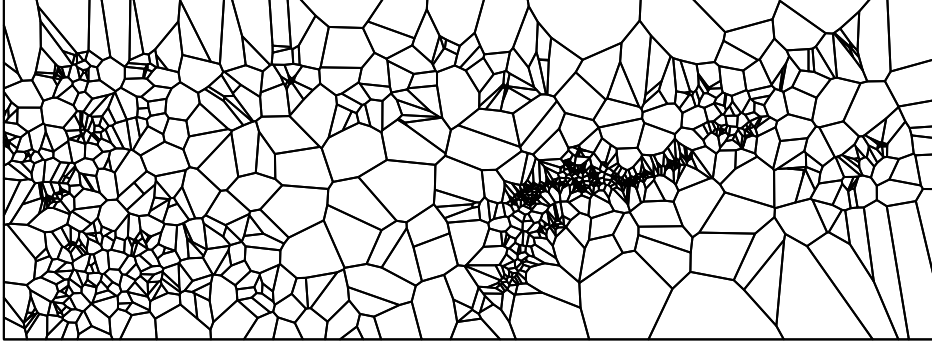
```
a_sf = st_as_sf(a, coords = c("Boylam", "Enlem"), crs = 4326)
a_sp = as(a_sf, "Spatial")
win = owin(xrange = bbox(a_sp)[1,], yrange = bbox(a_sp)[2,])
depem_ppp = ppp(x=coordinates(a_sp)[,1], y=coordinates(a_sp)[,2], window=win)
Q = quadratcount(depem_ppp, nx=20, ny=10)
plot(Q)
```

Q

0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
1	9	3	5	3	2	6	1	3	7	3	6	1	0	1	2	0	6	9	1
3	23	3	10	5	4	30	4	4	1	26	1	2	4	10	11	7	2	0	1
7	14	13	4	3	0	1	5	6	0	0	2	0	1	2	40	8	1	1	3
61	12	6	2	2	4	0	1	0	0	2	3	45	4	45	14	2	4	26	3
19	26	4	5	1	1	0	0	1	0	11	11	0	12	7	8	1	0	8	8
0	8	3	5	4	4	0	2	0	0	16	39	3	1	0	1	0	3	2	2
4	25	16	14	9	2	1	0	2	2	10	14	1	4	0	1	3	1	0	0
0	16	13	11	6	8	2	2	0	1	22	5	0	0	0	0	0	0	0	0
12	12	3	1	0	3	3	3	1	2	5	0	0	0	0	1	0	0	0	0

```
voronoi = dirichlet(depem_ppp)
plot(voronoi)
```

voronoi



Yoğun Bölgeler: Diyagramdaki bazı bölgelerde hücrelerin daha küçük ve daha sıkışık olduğunu görülmektedir. Bu, bu bölgelerdeki jeneratör noktalarının birbirine daha yakın olduğu anlamına gelir.

Seyrek Bölgeler: Diğer bölgelerde ise hücrelerin daha büyük ve daha seyrek olduğu gözlemlenmiştir. Bu da, bu bölgelerdeki jeneratör noktalarının birbirine daha uzak olduğu anlamına gelir.

```
veri = data.frame(  
row= c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,  
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,  
3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,  
4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,  
5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,  
6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,  
7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,  
8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,  
9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,  
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10),  
col= c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,  
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)  
)
```

```
komsular = cell2nb(max(veri$row), max(veri$col), type = "queen")  
agirliklar = nb2listw(komsular, style = "W")  
  
df_Q = as.data.frame(Q)  
moran_test = moran.test(df_Q$Freq, agirliklar)  
print(moran_test)
```

##

```
## Moran I test under randomisation
##
## data: df_Q$Freq
## weights: agirliklar
##
## Moran I statistic standard deviate = 4.4993, p-value = 3.408e-06
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.149592137      -0.005025126      0.001180909
```

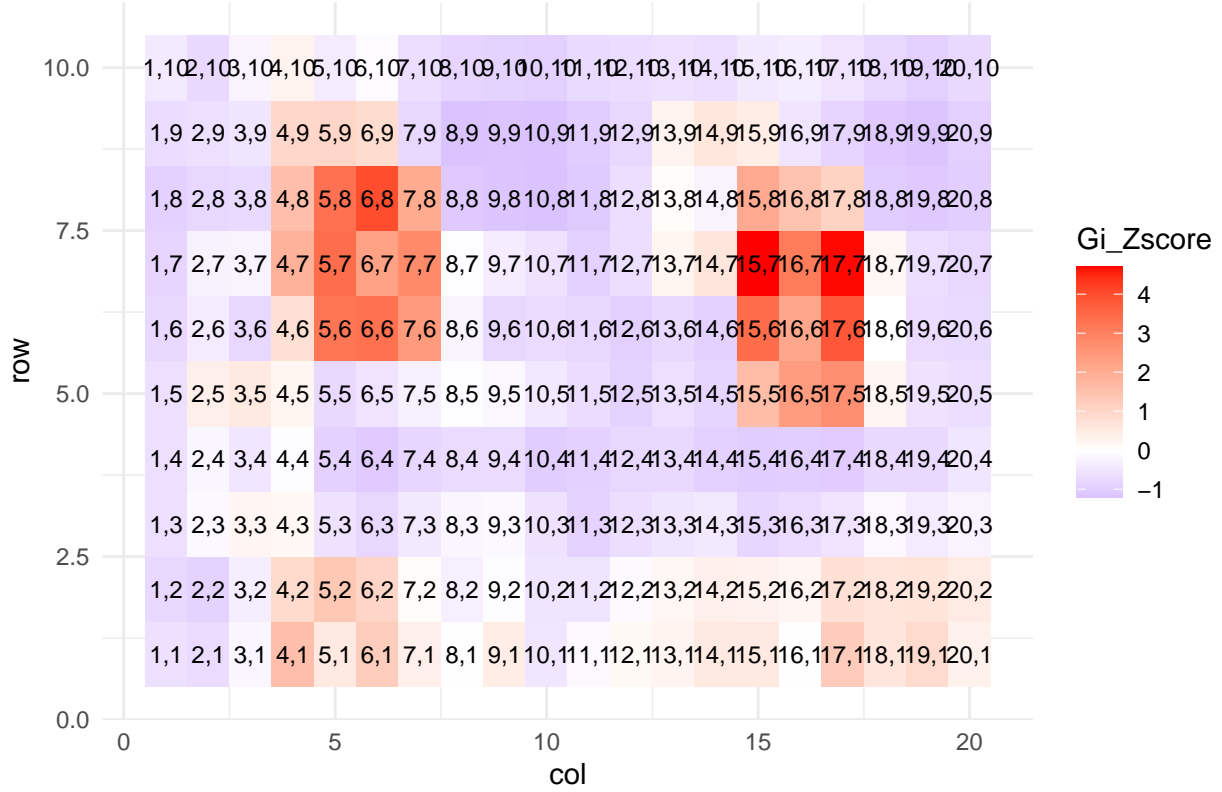
Hotspot Grafiği

```
gi_test = localG(df_Q$Freq, agirliklar)
veri$Gi_Zscore = gi_test
```

```
gi_test = localG(as.numeric(df_Q$Freq), agirliklar)
veri$Gi_Zscore = as.numeric(gi_test)

ggplot(veri, aes(x = col, y = row, fill = Gi_Zscore)) +
  geom_tile() +
  geom_text(aes(label = paste(col, row, sep = ",")), size = 3, color = "black") +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  ggtitle("Hotspot Analizi (Getis-Ord Gi*)") +
  theme_minimal()
```


Hotspot Analizi (Getis-Ord Gi*)

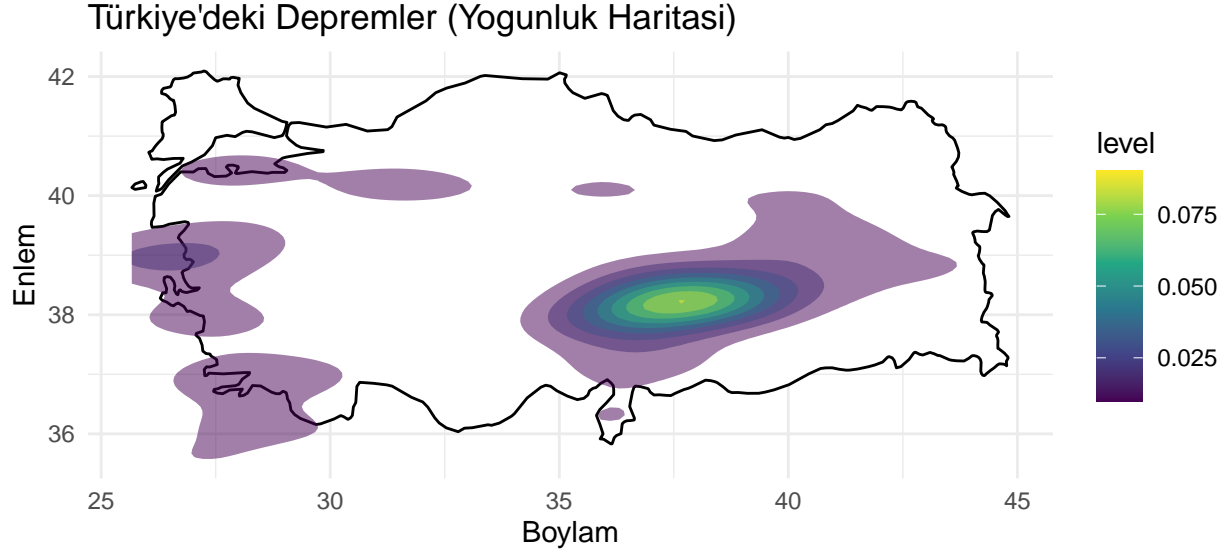


Kırmızı ve koyu kırmızı hücreler (yüksek pozitif Gi_Zscore): Bu bölgeler, istatistiksel olarak anlamlı derecede yüksek değerlerin kümelendiği (hotspot) alanları gösterir. Yani, bu konumlarda deprem aktivitesi yoğunluk çevresine göre belirgin şekilde daha yüksektir. Özellikle koyu kırmızı alanlar, en güçlü hotspotları işaret eder.

Yoğunluk Haritası(KDE)

```
ggplot() +
  geom_polygon(data = turkey_map, aes(x = long, y = lat, group = group), fill = "white", color = "black") +
  stat_density_2d(data = a, aes(x = Boylam, y = Enlem, fill = ..level..), geom = "polygon", alpha = 0.5) +
  scale_fill_viridis_c() +
  coord_fixed(1.3) +
  theme_minimal() +
  labs(title = "Türkiye'deki Depremler (Yoğunluk Haritası)", x = "Boylam", y = "Enlem")
```

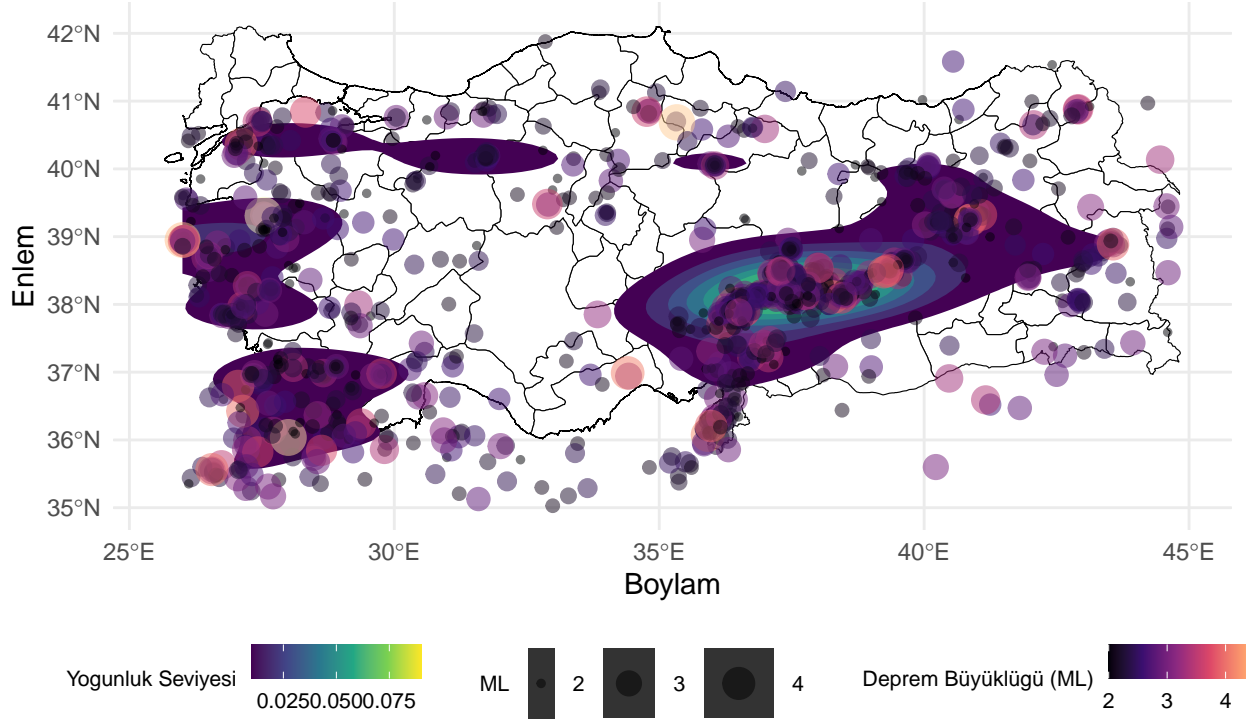
```
## Warning: The dot-dot notation (`..level..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(level)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Yukarıdaki yoğunluk haritası incelendiğinde, **Doğu Anadolu Bölgesi'nde**, özellikle **Kahramanmaraş**, **Malatya** ve **Adıyaman** çevresinde deprem yoğunluğunun belirgin şekilde fazla olduğu gözlemlenmektedir. Bu bölgelerdeki yüksek yoğunluk, deprem aktivitesinin yoğun olduğu alanları vurgulamaktadır.

```
ggplot() +  
  geom_sf(data = turkiye_il_sf, fill = "white", color = "black") +  
  stat_density2d(  
    data = a,  
    aes(x = Boylam, y = Enlem, fill = ..level..),  
    geom = "polygon",  
    alpha = 1  
  ) +  
  scale_fill_viridis_c(name = "Yoğunluk Seviyesi") +  
  geom_point(  
    data = a,  
    aes(x = Boylam, y = Enlem, color = ML, size = ML),  
    alpha = 0.5  
  ) +  
  scale_color_viridis_c(option = "magma", name = "Deprem Büyüklüğü (ML)") +  
  labs(  
    title = "Türkiye Üzerinde Deprem Yoğunluk ve Büyüklük Haritası",  
    x = "Boylam",  
    y = "Enlem"  
  ) +  
  theme_minimal() +  
  theme(  
    legend.key.size = unit(0.45, "cm"),  
    legend.text = element_text(size = 8),  
    legend.title = element_text(size = 8),  
    legend.position = "bottom"  
  )
```

Türkiye Üzerinde Deprem Yogunluk ve Büyüklük Haritası



Silhouette Yöntemi

Silhouette yöntemi, veri noktalarının kendi kümelerine ne kadar benzer olduğunu (cohesion) ve diğer kümelere ne kadar farklı olduğunu (separation) ölçen bir tekniktir. Bu yöntem, her bir veri noktası için bir silüet katsayısı hesaplayarak, kümelenebilirliğin kalitesini değerlendirir. Silüet katsayısı, bir veri noktasının doğru kümeye yerleştirilip yerleştirilmediğine dair önemli bir gösterge sunar.

$$f(x) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

şeklinde hesaplanır. Burada;

s(i): i veri noktasının silüet katsayısıdır. Bu değer -1 ile +1 arasında değişir.

a(i): i veri noktasının kendi kümesindeki diğer tüm veri noktalarına olan ortalama uzaklığıdır (küme içi uyumu ölçer).

b(i): i veri noktasının, ait olmadığı en yakın kümeye olan ortalama uzaklığıdır (kümeler arası ayrımı ölçer).

```
konumlar = a[, c("Boylam", "Enlem")]

k_list = 2:10
silhouette_skorlari = c()

for (k in k_list) {

  kmeans_result = kmeans(konumlar, centers = k)
```

```

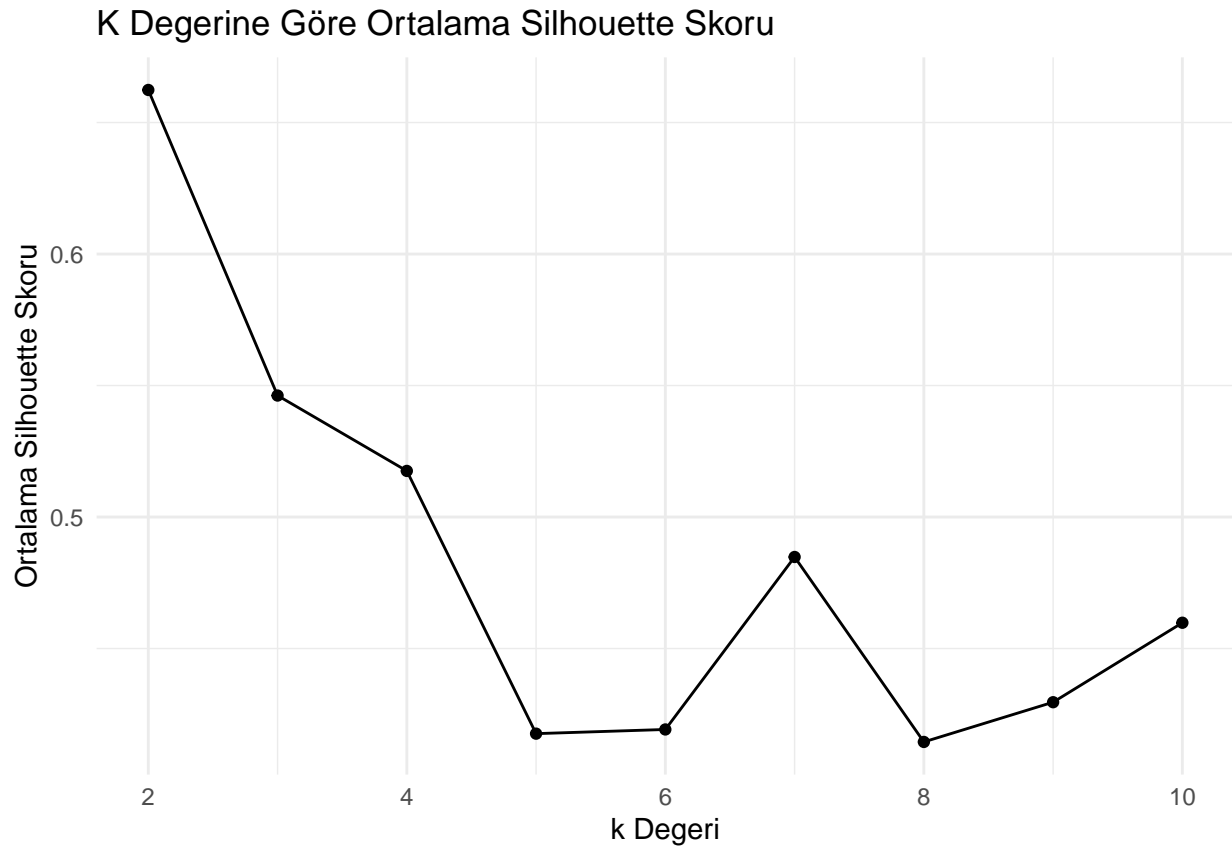
dist_mat = distm(konumlar, fun = distHaversine)
sil = silhouette(kmeans_result$cluster, dist_mat)
ort_sil = mean(sil[, "sil_width"])

silhouette_skorlari = c(silhouette_skorlari, ort_sil)
}

silhouette_df = data.frame(k = k_list, silhouette = silhouette_skorlari)

ggplot(silhouette_df, aes(x = k, y = silhouette)) +
  geom_line() + geom_point() +
  labs(title = "K Değerine Göre Ortalama Silhouette Skoru", x = "k Değeri", y = "Ortalama Silhouette Skoru") +
  theme_minimal()

```



`dist()` fonksiyonu, varsayılan olarak Öklidyen mesafesi kullanır, yani düz bir yüzey üzerinde iki nokta arasındaki doğrusal mesafeyi hesaplar. Ancak, analiz alanımız geniş bir coğrafi bölgeyi kapsıyorsa, `geosphere::distm()` fonksiyonunu kullanarak Büyük Çember Mesafesi (Haversine Distance) hesaplamayı düşünebiliriz. Bu yöntem, özellikle küresel yüzeydeki iki nokta arasındaki mesafeyi daha doğru bir şekilde belirler.

Yapılan analiz sonucunda, k değeri için en uygun değer 2 olduğu söylenebilir. Bu, kümelenmiş veri seti üzerinde elde edilen sonuçlara göre en iyi performansı gösteren parametredir.

Komsuluk Testleri

Moran I Testi

H_0 : Analiz edilen değişkende mekansal otokorelasyon yoktur. H_1 : Analiz edilen değişkende mekansal otokorelasyon vardır.

```
coords = st_coordinates(a_sf)
knn_neigh = knearneigh(coords, k=2)
knn_nb = knn2nb(knn_neigh)

## Warning in knn2nb(knn_neigh): neighbour object has 95 sub-graphs

listw = nb2listw(knn_nb, style="W")

moran_test = moran.test(a_sf$ML, listw)
print(moran_test)

##
## Moran I test under randomisation
##
## data: a_sf$ML
## weights: listw
##
## Moran I statistic standard deviate = 2.2197, p-value = 0.01322
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##      0.0544867597      -0.0007610350      0.0006194785
```

Analiz edilen değişkende, %95 güven düzeyinde istatistiksel olarak anlamlı pozitif mekansal otokorelasyon olduğu söylenebilir.

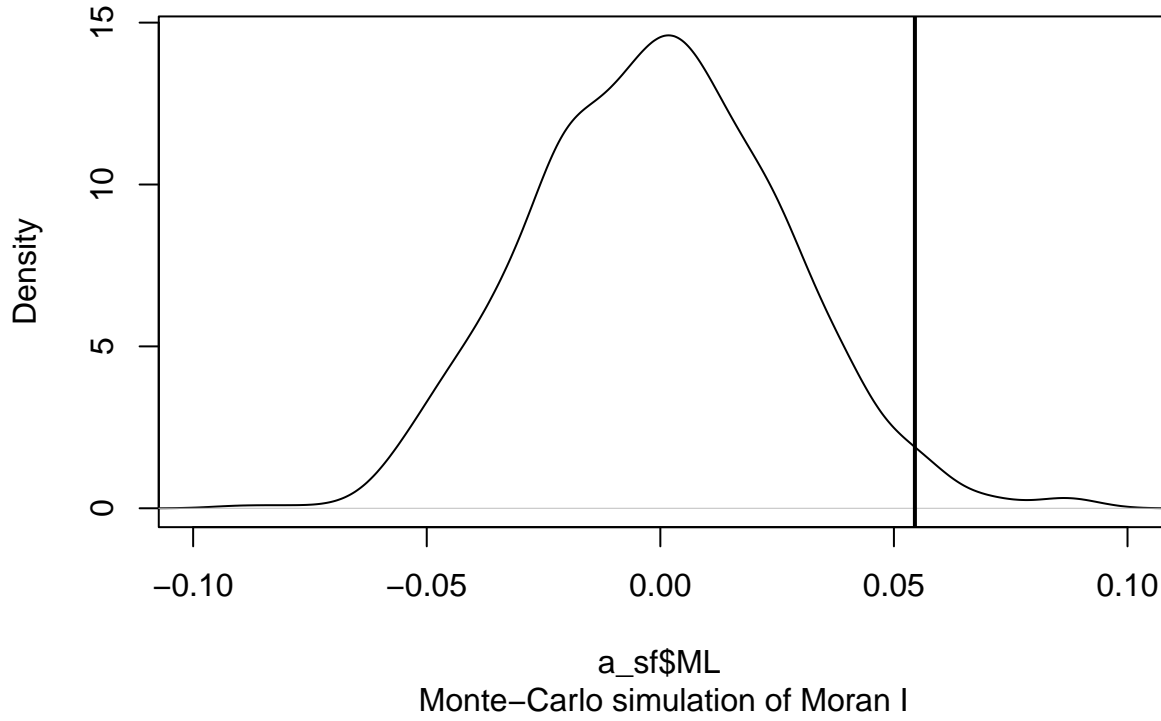
(Moran's I = 0.054, $p < 0.05$).

Bu sonuç, yüksek değerlerin birbirine yakın olma eğiliminde olduğunu ve düşük değerlerin de birbirine yakın gruplaştığını göstermektedir. Yani, mekansal dağılımda benzer değerlerin kümelenmesi söz konusudur.

```
moran_perm = moran.mc(a_sf$ML, listw, nsim = 999)

plot(moran_perm, main = "Monte Carlo Permutasyonları ile Moran's I Dağılımı")
```

Monte Carlo Permütasyonlari ile Moran's I Dagilimi

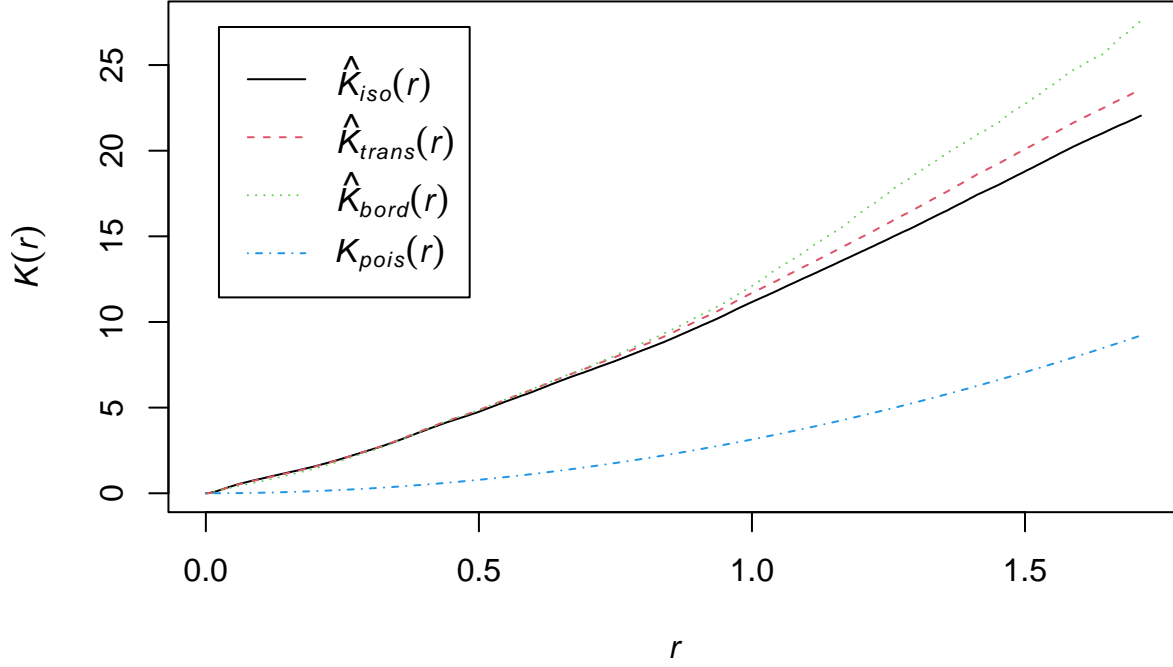


Grafikteki siyah çizgi, gözlemlenen Moran's I değerini temsil etmektedir. Dağılım ise, rastgele permütasyonlar ile elde edilen Moran's I değerlerinin gösterimidir. Eğer siyah çizgi, dağılımın uç noktalarına yakın bir yerde yer alıyorsa, bu durum sonucun istatistiksel olarak anlamlı olduğunu gösterir.

Ripley K

```
a_sf = st_as_sf(a, coords = c("Boylam", "Enlem"), crs = 4326)
a_sp = as(a_sf, "Spatial")
win = owin(xrange = bbox(a_sp)[1,], yrange = bbox(a_sp)[2,])
depren_ppp = ppp(x=coordinates(a_sp)[,1], y=coordinates(a_sp)[,2], window=win)
K_result = Kest(depren_ppp)
plot(K_result)
```

K_result



Kenar Düzeltme Yöntemlerinin Karşılaştırılması

Farklı kenar düzeltme yöntemlerinin sonuçları arasındaki tutarlılık, analizinizin robustluğunu değerlendirmek için önemli bir göstergedir. Eğer kenar düzeltme sonuçları arasında çizgiler birbirine yakınsa, bu durum kenar etkilerinin çok büyük bir sorun oluşturmadığını ve analiz sonuçlarının güvenilir olduğunu işaret edebilir.

Kümelenme ve Yayılma Ölçeklerinin İncelenmesi

Grafikte yer alan gözlemlenen K fonksiyonu eğrileri (kenar düzeltmeli) ile K_{pois} çizgisinin ne zaman ve ne kadar saptığına bakarak, verideki kümelenme veya yayılma özelliklerinin hangi mesafelerde (ölçeklerde) daha belirgin olduğunu anlayabiliriz.

LISA Testi

```
local_moran = localmoran(a_sf$ML, listw)
```

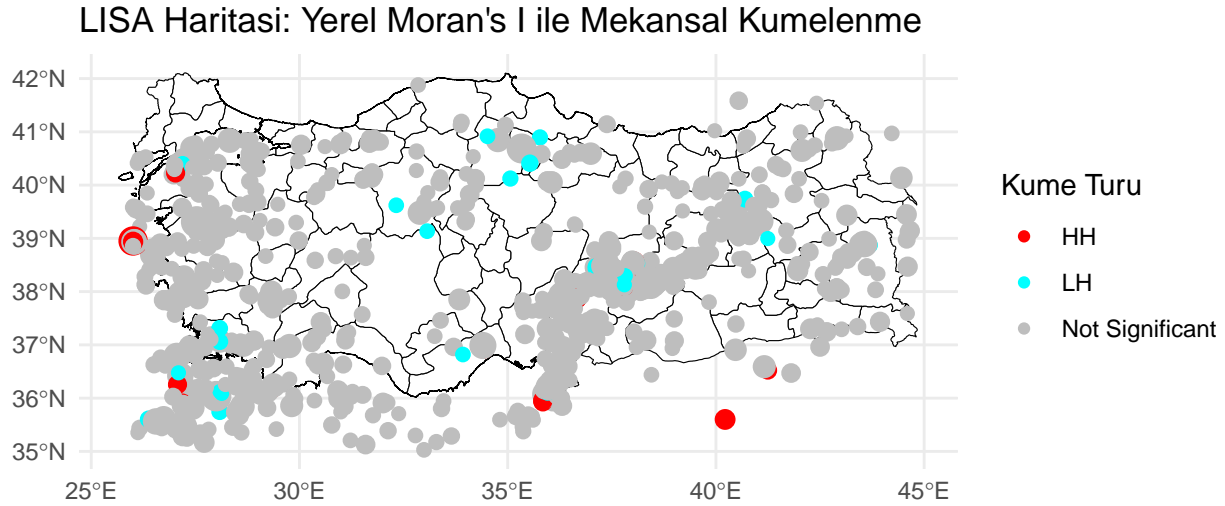
```
a_sf$lisa_I = local_moran[, "Ii"]
a_sf$lisa_p = local_moran[, "Pr(z != E(Ii))"]
```

```
a_sf$cluster = "Not Significant"
a_sf$cluster[a_sf$lisa_I > 0 & a_sf$ML > mean(a_sf$ML) & a_sf$lisa_p < 0.05] = "HH"
a_sf$cluster[a_sf$lisa_I > 0 & a_sf$ML < mean(a_sf$ML) & a_sf$lisa_p < 0.05] = "LL"
a_sf$cluster[a_sf$lisa_I < 0 & a_sf$ML > mean(a_sf$ML) & a_sf$lisa_p < 0.05] = "HL"
a_sf$cluster[a_sf$lisa_I < 0 & a_sf$ML < mean(a_sf$ML) & a_sf$lisa_p < 0.05] = "LH"
```

```
library(ggplot2)
ggplot(a_sf) +
```

```
geom_sf(data = turkiye_il_sf, fill = "white", color = "black") +
geom_sf(aes(color = cluster), size = a_sf$ML) +
scale_color_manual(values = c("HH" = "red", "LL" = "blue",
                              "HL" = "orange", "LH" = "cyan",
                              "Not Significant" = "grey")) +

theme_minimal() +
labs(title = "LISA Haritası: Yerel Moran's I ile Mekansal Kumelenme",
     color = "Kume Turu")
```



HH Bölgeleri ve Büyük Depremler

Haritada yer alan HH(High-High - Yüksek-Yüksek) bölgeleri, büyük depremlerin mekansal olarak yoğunlaştığı alanları temsil etmektedir. Bu bölgelerde meydana gelen depremlerin büyüklükleri, Türkiye genelindeki ortalama deprem büyüklüğünden belirgin şekilde daha yüksektir. Ayrıca, bu büyük depremler, birbirine yakın konumlarda (komşu olarak tanımlanan mesafede veya komşu sayısında) meydana gelme eğilimindedir. Yani, HH bölgeleri, ML değeri büyük depremlerin kümelendiği alanlardır.

LH Bölgeleri ve Küçük Depremler

LH (Low-High - Düşük - Yüksek) bölgeleri, küçük büyüklükteki bir deprem noktasının, komşuluk matrisine göre tanımlanan yakınındaki diğer deprem noktaları tarafından çevrelediği, bu komşu depremlerin ise yüksek büyüklüklere sahip olduğu alanlardır. Bu, küçük bir depremin, büyüklükleri genel ortalamasının üzerinde olan depremlerle çevrili olduğunu gösterir. Yani, LH bölgeleri, küçük büyüklükteki depremlerin büyük büyüklükteki depremlerle çevrelediği yerlerdir.

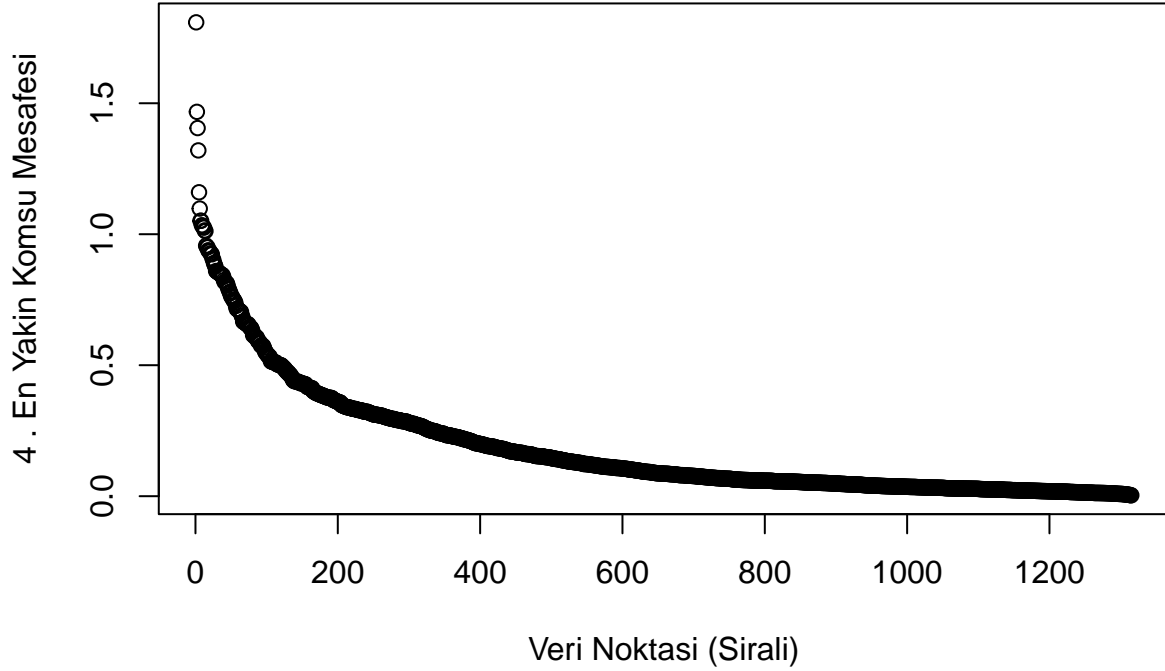
K-Distance Testi

```
k_degeri = 4

k_mesafeleri = knn.dist(konumlar, k = k_degeri)
siralanmis_mesafeler = sort(k_mesafeleri[, k_degeri], decreasing = TRUE)

plot(siralanmis_mesafeler,
     xlab = "Veri Noktası (Sıralı)",
     ylab = paste(k_degeri, ". En Yakın Komşu Mesafesi"),
     main = "K-distance Grafiği")
```


K-distance Grafiği



“Moran I testi ile depremlerin mekansal olarak kümелendiğini gözlemlemiştik. Şimdi ise bu kümelerin yapısını daha detaylı anlamak için bir yoğunluk tabanlı kümeleme algoritması olan DBSCAN’i uygulayacağız. DBSCAN algoritmasının temel parametrelerinden biri olan ‘eps’ (epsilon), bir noktanın çekirdek nokta olarak kabul edilebilmesi için etrafındaki maksimum komşu mesafesini ifade eder. Doğru ‘eps’ değerinin seçilmesi, anlamlı kümelerin ortaya çıkarılması açısından kritik öneme sahiptir.”

“K-distance grafiği, veri setimizdeki her bir nokta için 4. en yakın komşusuna olan mesafeyi sıralı bir şekilde göstermektedir. Grafikteki dikey eksen ‘4. En Yakın Komşu Mesafesi’ni, yatay eksen ise sıralanmış ‘Veri Noktaları’ni temsil etmektedir.”

“Grafiğin genel eğilimi, veri noktaları sıralandıkça 4. en yakın komşu mesafesinin azaldığını göstermektedir. Bu, veri setimizde genel olarak bir yoğunlaşma eğiliminin olduğunu düşündürmektedir. Ancak, özellikle grafiğin sol tarafında daha dik bir düşüş ve ardından daha yatay bir seyir gözlemlemekteyiz. İşte bu ‘dirsek noktası’, yoğun kümelerdeki noktaların birbirine olan mesafelerinin nispeten küçük olduğunu ve bu noktadan sonra mesafelerin daha hızlı arttığını, yani yoğun olmayan bölgelere geçildiğini işaret etmektedir.”

“eps değerini 0.6 olarak seçmemizin nedeni, bu grafikte gözlemlediğimiz dirsek noktasına yakın bir değer olmasıdır. Bu seçim, algoritmanın yaklaşık olarak 0.6 birim mesafe içinde kalan komşulukları yoğun kümeler olarak değerlendireceği anlamına gelir. Başka bir deyişle, bu mesafeye kadar olan noktaların, aynı yoğun kümenin parçası olma olasılığı daha yüksektir. Dirsek noktasının konumu, veri setimizdeki yoğun ve seyrek bölgeler arasındaki potansiyel sınırı göstermesi açısından önemlidir.”

DBSCAN Kümeleme Analizi

```
dbscan_result = dbscan(coords, eps = 0.6, minPts = 5)

a_sf$cluster = as.factor(dbscan_result$cluster)

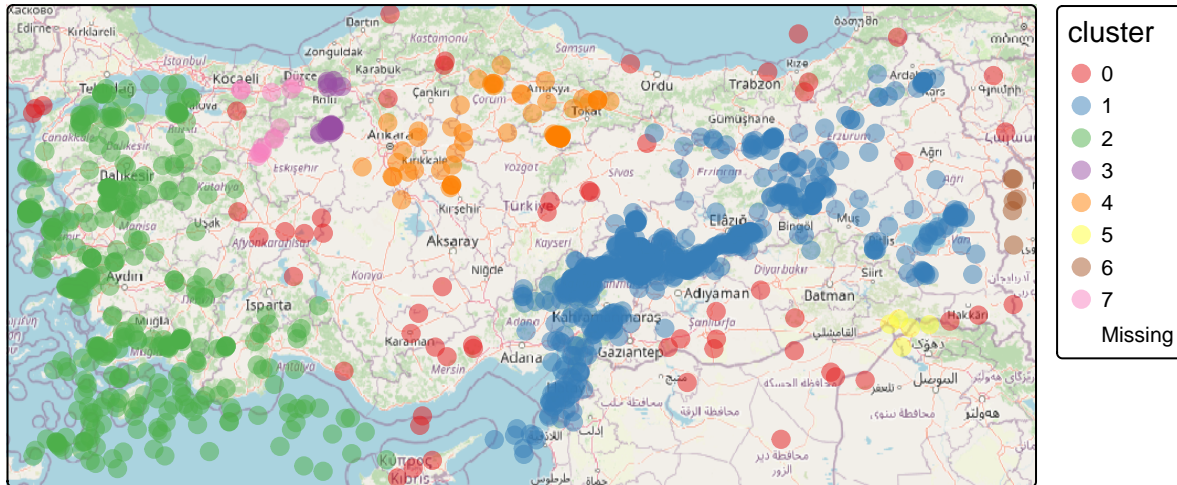
tm_shape(a_sf) +
```

```
tm_dots(col = "cluster", palette = "Set1", size=0.6,alpha=0.5) +
tm_basemap("OpenStreetMap")
```

```
##
```

```
## -- tmap v3 code detected -----
```

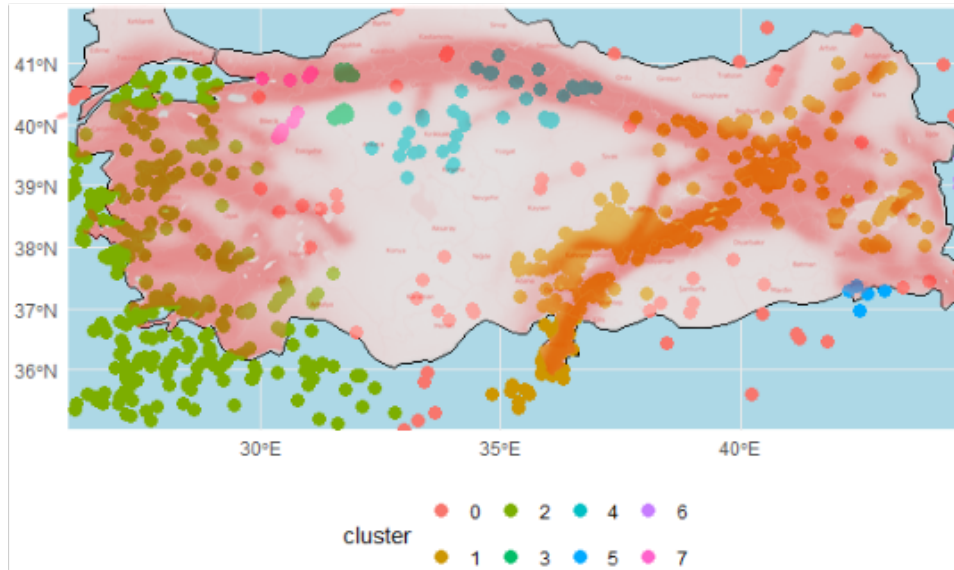
```
## [v3->v4] `tm_tm_dots()`: migrate the argument(s) related to the scale of the
## visual variable `fill` namely 'palette' (rename to 'values') to fill.scale =
## tm_scale(<HERE>).
## [v3->v4] `tm_dots()`: use 'fill' for the fill color of polygons/symbols
## (instead of 'col'), and 'col' for the outlines (instead of 'border.col').
## [v3->v4] `tm_dots()`: use `fill_alpha` instead of `alpha`.
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "Set1" is named
## "brewer.set1"
```



DBSCAN (Density-Based Spatial Clustering of Applications with Noise), yoğunluğa dayalı bir kümeleme algoritmasıdır. Temel amacı, veri noktalarını yoğun bölgeler halinde gruplandırmak ve seyrek bölgelerde tek başına duran noktaları “gürültü” olarak işaretlemektir. Yoğun Bölgeler Küme Oluşturur.

Bir noktanın etrafında belirli bir yarıçap (eps) içinde yeterli sayıda komşu nokta (MinPts) varsa, bu nokta bir kümenin çekirdeğini oluşturur ve komşuları da aynı kümeye dahil edilir.

Gürültü Tespiti: Düşük yoğunluklu bölgelerde bulunan ve hiçbir kümenin çekirdeğine yakın olmayan noktalar “gürültü” olarak etiketlenir (genellikle 0 değeri ile). Bu, aykırı değerleri (outlier) belirlemede faydalıdır.



Büyükölüklerine Göre Hotspot Grafiđi

```
gi_star = localG(a$ML, listw)

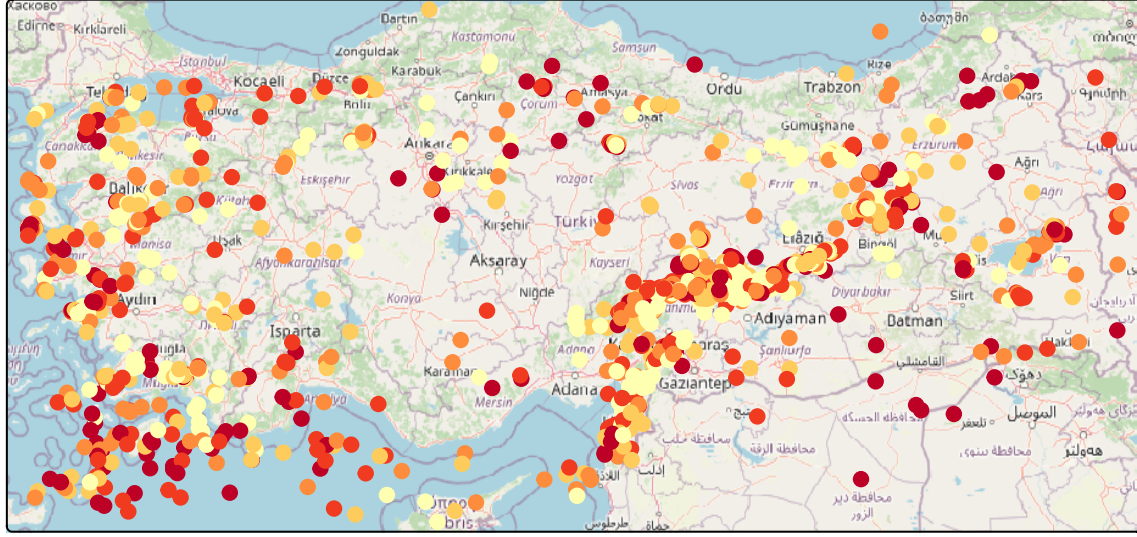
a_sf$Gi_star = as.numeric(gi_star)

tm_shape(a_sf) +
  tm_dots(col = "Gi_star", style = "quantile", palette = "YlOrRd", size=0.5) +
  tm_basemap("OpenStreetMap")

##

## -- tmap v3 code detected -----

## [v3->v4] `tm_dots()`: instead of `style = "quantile"`, use fill.scale =
## `tm_scale_intervals()`.
## i Migrate the argument(s) 'style', 'palette' (rename to 'values') to
## `tm_scale_intervals(<HERE>)'
## [cols4all] color palettes: use palettes from the R package cols4all. Run
## `cols4all::c4a_gui()` to explore them. The old palette name "YlOrRd" is named
## "brewer.yl_or_rd"
## Multiple palettes called "yl_or_rd" found: "brewer.yl_or_rd", "matplotlib.yl_or_rd". The first one,
```



Gi_star

- 1.367 to –0.880
- 0.880 to –0.395
- 0.395 to 0.090
- 0.090 to 0.899
- 0.899 to 4.949
- Missing

Bu analiz, belirli bir konumdaki deprem aktivitesinin, çevresindeki deprem aktivitesine göre istatistiksel olarak anlamlı derecede yüksek mi yoksa düşük mü olduğunu belirlememize olanak tanır.

Kırmızı ve koyu kırmızı noktalar, çevresine göre istatistiksel olarak anlamlı derecede ML değeri büyük deprem aktivitesinin (sıcak noktalar - hot spots) olduğu bölgeleri göstermektedir.

Kaynakça

Aral, N., & Aytaç, M. (2018). TÜRKİYE'DE İŞSİZLİĞİN MEKÂNSAL ANALİZİ. Öneri Dergisi, 13(49), 1-20. <https://doi.org/10.14783/maruoneri.vi.364680>

Bartan, M. B., Aytaç Cankurtaran, G., & Türkşen, Ö. (2025). Mekânsal Nokta Verilerin Mekânsal İstatistiklerle Analizi: Yenilenebilir Enerji Kaynakları Uygulaması. International Journal of Advances in Engineering and Pure Sciences, 37(UYIK 2024 Special Issue), 40-49. <https://doi.org/10.7240/jeps.1527355>

<https://dergipark.org.tr/en/download/article-file/1751876>

<https://builtin.com/data-science/elbow-method>

<https://www.youtube.com/watch?v=MV6dgAGTv-k&themeRefresh=1>

Webster, R., and P. Burrough. 1972. "Computer-Based Soil Mapping of Small Areas from Sample Data II. Classification Smoothing." Journal of Soil Science 23 (2): 222-34.

EKOLOJİ LABORATUAR KİTABI. 2006. POPULASYONLARIN ALANSAL YAPISI (Yunus.hacettepe.edu.tr) (Bölüm 4)