

T.C
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
Yönetim Bilişim Sistemleri



2023-2024
Sosyal Medya ve Web Analizi
Raporu
Marmara Depremi Emoji Analizi

Bilal YEŞİLTAŞ

Bilecik-2024

İçindekiler

- Giriş
- Özet
- R Programlama Kütüphaneleri
- R Programlama ile Youtubeden Veri Çekme Yöntemi
- Excel'deki Veri Setini R Programlamaya Yükleme ve Emojileri Çıkartma
- R Veri Çerçevesindeki Emojileri Temizleme ve Düzeltme İşlemleri
- Geçerli Emojileri Filtreleme Fonksiyonu
- Çubuk Grafiği
- Emoji Bulutu
- Duygu Analizi
- Emojilerin Anlamları ve Kullanım Sebepleri
- Sonuç
- Kaynakça

Giriş

Marmara Depremi, 17 Ağustos 1999'da Türkiye'nin kuzeybatısında, özellikle Marmara Bölgesi'nde meydana gelen ve 7.4 büyüklüğünde olan yıkıcı bir doğa olayıdır. Depremi merkezi, Kocaeli'nin Gölcük ilçesi olarak belirlenmiştir ve sarsıntı yaklaşık 37 saniye sürmüştür. Bu felaket, 17,000'den fazla insanın hayatını kaybetmesine, 25,000'den fazla insanın yaralanmasına ve binlerce binanın yıkılmasına veya hasar görmesine yol açmıştır. Marmara Depremi, sadece fiziksel yıkıma neden olmakla kalmamış, aynı zamanda büyük ekonomik kayıplara ve geniş çaplı göçlere neden olarak bölgedeki yaşamı derinden etkilemiştir.

Deprem sonrasında Türkiye, deprem yönetimi ve yapı denetimi konularında önemli adımlar atmış, bu süreçte uluslararası yardımların bölgeye ulaşması sağlanmıştır. Depremi etkisiyle birlikte, ülke genelinde yapı standartları gözden geçirilmiş ve yeni düzenlemeler getirilmiştir. Ayrıca, bu felaketin ardından toplumda deprem bilinci ve hazırlık düzeyi de önemli ölçüde artmıştır.

Marmara Depremi, bireylerin ve toplulukların bu tür felaketlere karşı nasıl tepki verdiği ve bu olayların duygusal etkilerinin nasıl yönetildiği konusunda da önemli bir örnek teşkil etmektedir. Özellikle sosyal medya platformlarında, insanların bu tür olaylara verdikleri tepkiler ve bu tepkilerin duygusal yansımaları, toplumun genel ruh halini ve olaylara bakış açısını anlamak için değerli veriler sunmaktadır.

Bu bağlamda, YouTube gibi geniş kitlelere ulaşan platformlarda paylaşılan içerikler ve bu içeriklere yapılan yorumlar, toplumsal tepkilerin analiz edilmesi açısından önemli bir kaynak olarak karşımıza çıkmaktadır. Bu araştırmanın temel amacı, Marmara Depremi ile ilgili YouTube videolarına yapılan yorumlardaki emojilerden elde edilen duyguları analiz ederek, bireylerin bu felakete yönelik duygusal tepkilerini anlamaktır.

ÖZET

Bu araştırmanın temel hedefi, YouTube platformunda Marmara Depremi ile ilgili paylaşılan videolara yapılan yorumlardaki emojilerden elde edilen duyguları ortaya çıkarmak ve bu yolla bireylerin duygusal yanıtlarını anlamaktır. Toplamda 19 farklı video üzerinden 20,132 adet Türkçe yorum toplanmıştır. Bu yorumlarda kullanılan emojiler ayıklanarak toplamda 2,043 emoji tespit edilmiştir. NA değerleri olan emojiler çıkarıldığında ise toplam 846 emoji elde edilmiştir. Veri seti, açık kaynak kodlu R programlama dili kullanılarak analiz edilmiştir. Gerçekleştirilen analiz, metin madenciliği yöntemlerini kullanarak Marmara Depremi'ne dair en sık tekrar edilen emojilerin belirlenmesini, emoji grafiğinin ve emoji bulutunun oluşturulmasını, ayrıca emoji duygu analizinin yapılmasını içermektedir.

Analiz sürecinde, öncelikle yorumlarda en sık kullanılan emojilerin belirlenmesi için frekans analizi yapılmıştır. Bu analiz sonucunda, yorumlarda en çok tekrar eden emojiler tespit edilmiştir. Elde edilen veriler kullanılarak bir emoji bulutu oluşturulmuş ve bu bulut üzerinden emojilerin görsel dağılımı incelenmiştir. Ayrıca, emojilerin duygusal içeriklerine göre sınıflandırılması yapılmış ve bu sınıflandırma üzerinden duygu analizi gerçekleştirilmiştir.

Marmara Depremi ile ilişkili emojilerin yüzde dağılımına baktığımızda, çoğunlukla negatif duyguların hakim olduğu görülmektedir. Üzüntü ve endişe gibi negatif duygular %61'ini oluştururken, pozitif duygular %39'luk bir dilimi kapsamaktadır. Bu analiz, sosyal medya platformlarında Marmara Depremi'ne yönelik duygusal tepkilerin büyük ölçüde olumsuz olduğunu, ancak aynı zamanda toplumsal dayanışma ve destek duygularının da var olduğunu göstermektedir. Yorumlarda gözlemlenen bu duygusal tepkiler, deprem gibi büyük felaketlerin toplum üzerindeki derin etkilerini anlamak ve bu tür olaylara yönelik duygusal yanıtları daha iyi yönetmek için önemli ipuçları sunmaktadır.

Sonuç olarak, bu araştırma Marmara Depremi'ne yönelik toplumsal duygusal tepkileri anlamak için değerli bir veri seti sunmakta ve bu tepkilerin analiz edilmesi, benzer durumlarla başa çıkmak için stratejilerin geliştirilmesine katkıda bulunmaktadır. Bu tür analizler, gelecekteki felaket yönetimi ve toplumsal dayanışma stratejilerinin daha etkili bir şekilde planlanmasına yardımcı olabilir.

R Programlamada Kullandığım

Kütüphaneler

R programlama dilinde, YouTube uygulamasından veri çekip emoji analiz etmek için kullanılan hazır kütüphaneleri indirmek için "install.packages()" komutu kullanılır

```
```{r}
install.packages("googlesheets4")
install.packages("httr")
install.packages("readxl")
install.packages("googleAuthR")
install.packages("tuber")
install.packages("stringi")
install.packages("stringr")
install.packages("dplyr")
install.packages("showtext")
install.packages("ggplot2")
install.packages("wordcloud2")
install.packages("tidytext")
```
```

R programlama dilinde, YouTube uygulamasından veri çekip emoji analiz etmek için kullanılan hazır kütüphaneleri etkinleştirmek için "library()" komutları kullanılır. Aşağıda, bu kütüphaneleri etkinleştirmek için kullanılan komutlar yer almaktadır.

```
```{r}
library(googlesheets4) # Google Sheets ile R arasında veri alışverişi yapmanızı sağlar.
library(httr) # HTTP istekleri yapmak ve yanıtları işlemek için kullanılır.
library(readxl) # Excel dosyalarını R ortamına aktarmanızı sağlar.
library(googleAuthR) # Google API'leriyle etkileşime geçmek için kimlik doğrulaması yapmanızı sağlar.
library(tuber) # YouTube API'leriyle etkileşime geçmek için kullanılır.
library(stringi) # Dizeler üzerinde işlem yapmak için geniş bir fonksiyon koleksiyonu sağlar.
library(stringr) # Dizeler üzerinde daha kullanıcı dostu işlemler yapmanızı sağlar.
library(dplyr) # Veri manipülasyonu ve filtreleme için kullanılır.
library(showtext) # Grafiklerde farklı yazı tiplerini kullanmanızı sağlar.
library(ggplot2) # Verileri görselleştirmek için kullanılır.
library(wordcloud2) # Kelime bulutları oluşturmaya olanak sağlar.
library(tidytext) # Metin madenciliği ve duygu analizi gibi işlemleri gerçekleştirmek için kullanılır.
```
```

R Programlama ile Youtubeden Veri Çekme Yöntemi

R programlama dilinde, kütüphanelerin etkinleştirilmesi sonrasında düzenli olarak alınan veriler bir araya getirilmiş ve bu veriler, R ortamına çağrıldığı şekillerde aşağıda gösterildiği gibi ifade edilmiştir.

```
{r}
client_id <- "*****"
client_secret <- "*****"
yt_oauth(client_id, client_secret, token = '')
get_all_comments(video_id = "*****")
comment <- get_all_comments (video_id = "*****")
write_xlsx (comment, "C:\\Youtube\\marmaradepremi.xlsx")
```

client_id ve client_secret: Bu değişkenler, YouTube API ile iletişim kurmak için kullanılacak olan uygulama kimlik bilgilerini içerir. client_id, uygulamanın tanımlayıcısıdır, ve client_secret ise uygulamanın gizli anahtarını içerir.

yt_oauth(client_id, client_secret, token = ""): Bu satır, YouTube API'ye yetki sağlamak için OAuth kimlik doğrulamasını gerçekleştirir. yt_oauth fonksiyonu, belirtilen client_id ve client_secret ile yetkilendirme yapar. token = " parametresi, mevcut bir erişim belirteci (token) kullanılmadığını belirtir.

get_all_comments(video_id = "Mc8ts57Za4w"): Bu fonksiyon, belirtilen video ID'sine ait tüm yorumları çeker. video_id parametresi, hedef videoyu tanımlar. Bu örnekte, "Mc8ts57Za4w" olarak belirtilmiştir.

comment <- get_all_comments(video_id = "Mc8ts57Za4w"): Bu satır, çekilen tüm yorumları comment adlı bir değişkene atar.

write_xlsx(comment, "C:\\Youtube\\marmaradepremi.xlsx"): Bu satır, comment değişkenindeki yorumları "C:\\Youtube\\yorumlar12.xlsx" adlı Excel dosyasına kaydeder. write_xlsx fonksiyonu, bir veri çerçevesini Excel dosyasına yazmak için kullanılır.

Bu kod dizisi, R programlama dilinde YouTube API'sini kullanarak belirli bir videoya ait tüm yorumları çekmek ve bu yorumları bir Excel dosyasına kaydetmek amacıyla kullanılmıştır.

Excel'deki Veri Setini R Programlamaya Yükleme ve Emojileri Çıkartma

Veri çekme işlemi tamamlandıktan sonra, Excel'e aktardığımız verileri okuyalım.

```
## Veri çekme işlemi tamamlandıktan sonra, excele aktardığımız verileri okuyalım.
{r}
deprem <- read_excel(file.choose(), sheet = 1)
```

```
deprem <- read_excel(file.choose(), sheet = 1)
```

Açıklama:

- read_excel fonksiyonu, Excel dosyasından veri okumak için kullanılır. Bu fonksiyon, readxl paketinin bir parçasıdır.
- file.choose() fonksiyonu, dosya seçim penceresi açarak kullanıcıya dosya seçme imkanı verir.
- sheet = 1 parametresi, Excel dosyasının 1. sayfasını okuyacağımızı belirtir.
- Bu kod çalıştırıldığında, bir dosya seçme penceresi açılır ve seçilen dosyanın 1. sayfasındaki veriler deprem adlı veri çerçevesine yüklenir.

Metinde yer alan emojileri metinden çıkaralım.

```
## Metinde yer alan emojileri metinden çıkaralım.
{r}
emojiler <- stri_extract_all_regex(deprem$metin, "\\p{So}|\\p{Sk}")
```

```
emojiler <- stri_extract_all_regex(deprem$metin, "\\p{So}|\\p{Sk}")
```

Açıklama:

- stri_extract_all_regex fonksiyonu, düzenli ifadeler kullanarak bir metin içindeki belirli öğeleri çıkarmak için kullanılır. Bu fonksiyon, stringi paketinin bir parçasıdır.
- deprem\$metin ifadesi, deprem veri çerçevesindeki metin sütununu belirtir. Bu sütun, analiz edilen metinleri içerir.
- \\p{So}\\p{Sk} düzenli ifadeleri, Unicode karakter sınıflarını belirtir. \\p{So} diğer semboller (Other Symbols) ve \\p{Sk} sembol modifikasyonları (Symbol Modifiers) anlamına gelir. Bu, metin içindeki tüm emojileri ve sembol karakterlerini bulmak için kullanılır.
- Bu kod çalıştırıldığında, deprem\$metin sütunundaki tüm emojiler ve semboller emojiler adlı listeye çıkarılır.

Çıkartmış olduğumuz emojiplerden 'NA' değerlerini temizleyerek bir emoji listesi oluşturalım.

```
## Çıkartmış olduğumuz emojiplerden 'NA' değerlerini temizleyerek bir emoji listesi oluşturalım.
...{r}
temizlenmis_emojipler <- emojipler[!is.na(emojipler)]
...
```

```
temizlenmis_emojipler <- emojipler[!is.na(emojipler)]
```

Açıklama:

- emojipler listesi, bir önceki adımda çıkarılan emojipleri içerir.
- !is.na(emojipler) ifadesi, emojipler listesindeki NA (eksik) değerleri tespit eder. is.na fonksiyonu NA değerlerini TRUE olarak döner. ! operatörü bu değeri tersine çevirerek FALSE yapar.
- emojipler[!is.na(emojipler)] ifadesi, NA değerleri olmayan öğeleri seçer.
- Bu kod çalıştırıldığında, temizlenmis_emojipler adlı yeni bir liste oluşturulur ve bu liste NA değerleri içermeyen tüm emojipleri içerir.

R Veri Çerçevesindeki Emojileri Temizleme ve Düzeltme İşlemleri

Kaç farklı emojiinin kullanıldığını belirleyelim.

```
## Kaç farklı emojiinin kullanıldığını belirleyelim.
...{r}
kac_farkli_emoji_kullanilmis <- length(unique(temizlenmis_emojipler))
print(kac_farkli_emoji_kullanilmis)
...
```

```
kac_farkli_emoji_kullanilmis <- length(unique(temizlenmis_emojipler))
print(kac_farkli_emoji_kullanilmis)
```

Açıklama:

- unique fonksiyonu, bir vektördeki benzersiz öğeleri döner.
- length fonksiyonu, bir vektördeki öğelerin sayısını döner.
- Bu kod, benzersiz emojiplerin sayısını hesaplar ve ekrana yazdırır.

Aşağıdaki fonksiyon, showtext paketinin işlevselliğinin otomatik kullanımını açıp kapatabilir.

```
## Aşağıdaki fonksiyon, showtext paketinin işlevselliğinin otomatik kullanımını açıp kapatabilir.
...{r}
showtext_auto()
...
```

showtext_auto()

Açıklama:

- showtext_auto fonksiyonu, showtext paketinin işlevselliğini otomatik olarak etkinleştirir veya devre dışı bırakır. Bu, grafiklerdeki metinlerin daha iyi görselleştirilmesini sağlar.

Temizlenmiş tüm emojileri unlist() fonksiyonu ile düz bir vektöre dönüştürelim. Yani, bir liste içindeki tüm alt elemanları tek bir vektör haline getirelim.

```
## Temizlenmiş tüm emojileri unlist() fonksiyonu ile düz bir vektöre dönüştürelim.
Yani, bir liste içindeki tüm alt elemanları tek bir vektör haline getirelim.
...{r}
tum_emojiler <- unlist(temizlenmis_emojiler)
...
```

tum_emojiler <- unlist(temizlenmis_emojiler)

Açıklama:

- unlist fonksiyonu, bir listeyi düz bir vektöre dönüştürür.
- Bu kod, temizlenmis_emojiler listesindeki tüm emojileri tek bir vektör haline getirir.

Bir emoji'nin frekansını (kaç kez geçtiğini) hesaplayalım. table() fonksiyonu ile bu frekansları bir tablo (vektör) olarak saklayalım.

```
## Bir emoji'nin frekansını (kaç kez geçtiğini) hesaplayalım.
table() fonksiyonu ile bu frekansları bir tablo (vektör) olarak saklayalım.
...{r}
emoji_freq <- table(tum_emojiler)
...
```

emoji_freq <- table(tum_emojiler)

Açıklama:

- table fonksiyonu, bir vektördeki öğelerin frekanslarını hesaplar.
- Bu kod, tum_emojiler vektöründeki her bir emoji'nin kaç kez geçtiğini hesaplar ve frekansları bir tablo (vektör) olarak saklar.

Saklanan tabloyu iki sütunlu bir veri çerçevesine dönüştürelim.

```
##Saklanan tabloyu iki sütunlu bir veri çerçevesine dönüştürelim.
{r}
emoji_freq_df <- data.frame(emoji = names(emoji_freq), frequency = as.numeric(emoji_freq))
```

```
emoji_freq_df <- data.frame(emoji = names(emoji_freq), frequency =
as.numeric(emoji_freq))
```

Açıklama:

- data.frame fonksiyonu, iki sütunlu bir veri çerçevesi oluşturur.
- names(emoji_freq) ifadesi, emoji_freq tablosundaki emoji adlarını alır.
- as.numeric(emoji_freq) ifadesi, emoji_freq tablosundaki frekansları sayısal değerlere dönüştürür.
- Bu kod, emojilerin ve frekanslarının yer aldığı iki sütunlu bir veri çerçevesi oluşturur.

str() fonksiyonu ile veri çerçevesinin özetini çıkaralım.

```
## str() fonksiyonu ile veri çerçevesinin özetini çıkaralım.
{r}
str(emoji_freq_df)
```

```
str(emoji_freq_df)
```

Açıklama:

- str fonksiyonu, bir veri çerçevesinin yapısını özetler.
- Bu kod, emoji_freq_df veri çerçevesinin yapısını ekrana yazdırır.

Geçerli Emojileri Filtreleme Fonksiyonu

```
{r}
is_valid_emoji <- function(emoji) {
  emoji_codepoints <- stri_enc_toutf32(emoji)
  if (length(emoji_codepoints) == 0) return(FALSE)

  emoji_ranges <- list(
    c(0x1F600, 0x1F64F), # Emoticons
    c(0x1F300, 0x1F5FF), # Miscellaneous Symbols and Pictographs
    c(0x1F680, 0x1F6FF), # Transport and Map Symbols
    c(0x1F700, 0x1F77F), # Alchemical Symbols
    c(0x1F780, 0x1F7FF), # Geometric Shapes Extended
    c(0x1F800, 0x1F8FF), # Supplemental Arrows-C
    c(0x1F900, 0x1F9FF), # Supplemental Symbols and Pictographs
    c(0x1FA00, 0x1FA6F), # Chess Symbols
    c(0x1FA70, 0x1FAFF) # Symbols and Pictographs Extended-A
  )

  any(sapply(emoji_ranges, function(range) {
    any(emoji_codepoints >= range[1] & emoji_codepoints <= range[2])
  }))
}
```

```

is_valid_emoji <- function(emoji) {
  emoji_codepoints <- stri_enc_toutf32(emoji)
  if (length(emoji_codepoints) == 0) return(FALSE)

  emoji_ranges <- list(
    c(0x1F600, 0x1F64F), # Emoticons
    c(0x1F300, 0x1F5FF), # Miscellaneous Symbols and Pictographs
    c(0x1F680, 0x1F6FF), # Transport and Map Symbols
    c(0x1F700, 0x1F77F), # Alchemical Symbols
    c(0x1F780, 0x1F7FF), # Geometric Shapes Extended
    c(0x1F800, 0x1F8FF), # Supplemental Arrows-C
    c(0x1F900, 0x1F9FF), # Supplemental Symbols and Pictographs
    c(0x1FA00, 0x1FA6F), # Chess Symbols
    c(0x1FA70, 0x1FAFF) # Symbols and Pictographs Extended-A
  )

  any(sapply(emoji_ranges, function(range) {
    any(emoji_codepoints >= range[1] & emoji_codepoints <= range[2])
  })))
}

```

Açıklama:

- is_valid_emoji fonksiyonu, bir emoji'nin geçerli olup olmadığını belirler.
- stri_enc_toutf32 fonksiyonu, bir emoji'yi Unicode kod noktasına dönüştürür.
- emoji_ranges listesi, geçerli emoji aralıklarını içerir.
- sapply fonksiyonu, her bir emoji aralığını kontrol eder ve emoji'nin bu aralıkta olup olmadığını belirler.
- Bu fonksiyon, geçerli emoji'ler için TRUE, geçerli olmayan emoji'ler için FALSE döner.

```

'''{r}
temizlenmis_emojiler <- temizlenmis_emojiler[sapply(temizlenmis_emojiler, is_valid_emoji)]
'''

temizlenmis_emojiler <- temizlenmis_emojiler[sapply(temizlenmis_emojiler,
is_valid_emoji)]

```

Açıklama:

- sapply fonksiyonu, temizlenmis_emojiler listesindeki her bir emoji'yi is_valid_emoji fonksiyonuna gönderir.
- Bu kod, geçerli emoji'lerden oluşan bir liste döner.

Çubuk Grafik - En Fazla Kullanılan 40 Emoji

```
## Veri çerçevesindeki emojiilerin frekanslarına göre sıralanmış ilk 40 emojiyi içeren bir altküme oluşturalım.  
{r}  
top_40_emojiler <- emoji_freq_df %>%  
  arrange(desc(frequency)) %>%  
  head(40)
```

```
top_40_emojiler <- emoji_freq_df %>%  
  arrange(desc(frequency)) %>%  
  head(40)
```

Açıklama:

- arrange() ve head() fonksiyonları kullanılarak emoji frekanslarına göre en fazla kullanılan 40 emoji top_40_emojiler veri çerçevesinde saklanır.

Top 40 Emojileri Seçme

```
## 40 emojiyi içeren bu altküme emoji sütununu seçerek yeni bir vektör oluşturalım.  
{r}  
top_40_emojiler_unicode <- top_40_emojiler$emoji
```

```
top_40_emojiler_unicode <- top_40_emojiler$emoji
```

Açıklama:

- top_40_emojiler veri çerçevesinden emoji sütununu seçerek, en sık kullanılan 40 emojiyi temsil eden top_40_emojiler_unicode adlı bir vektör oluşturulur.

Emojilerin Evrensel Kodlarını İşleme

```
## Emojilerin evrensel kodlarını işleyelim.  
{r}  
unicode_liste <- stri_extract_all_regex(top_40_emojiler_unicode, "[\\p{So}\\p{Sk}]+")
```

```
unicode_liste <- stri_extract_all_regex(top_40_emojiler_unicode, "[\\p{So}\\p{Sk}]+")
```

Açıklama:

- stri_extract_all_regex() fonksiyonuyla top_40_emojiler_unicode vektöründeki her bir emojiin Unicode kodunu unicode_liste adlı bir liste olarak çıkarırız.

Emoji Etiketlerini ve Frekanslarını Ayrı Ayrı Almak

```
## Emojilerin etiketlerini ve frekanslarını ayrı ayrı olarak hesaplayalım. Bunun için iki farklı vektör oluşturalım.
{r}
emoji_labels <- top_40_emojiler$emoji
emoji_counts <- top_40_emojiler$frequency
```

```
emoji_labels <- top_40_emojiler$emoji
emoji_counts <- top_40_emojiler$frequency
```

Açıklama:

- top_40_emojiler\$emoji: top_40_emojiler veri çerçevesindeki emoji sütunu, en sık kullanılan 40 emoji'nin Unicode karakterlerini içeren bir vektör olarak emoji_labels adıyla atanır.
- top_40_emojiler\$frequency: top_40_emojiler veri çerçevesindeki frequency sütunu, her bir emoji için kullanım sıklığını temsil eden bir vektör olarak emoji_counts adıyla atan

Veri Çerçevesindeki En Yüksek Frekansa Sahip İlk 40 Emojinin Unicode Karakterlerini İçeren Bir Vektör Oluşturma

```
## Veri çerçevesindeki "emoji" sütununu döndürerek en yüksek frekansa sahip ilk 40 emojinin Unicode karakterlerini içeren bir vektör oluşturalım.
{r}
top_40_emojiler$emoji
```

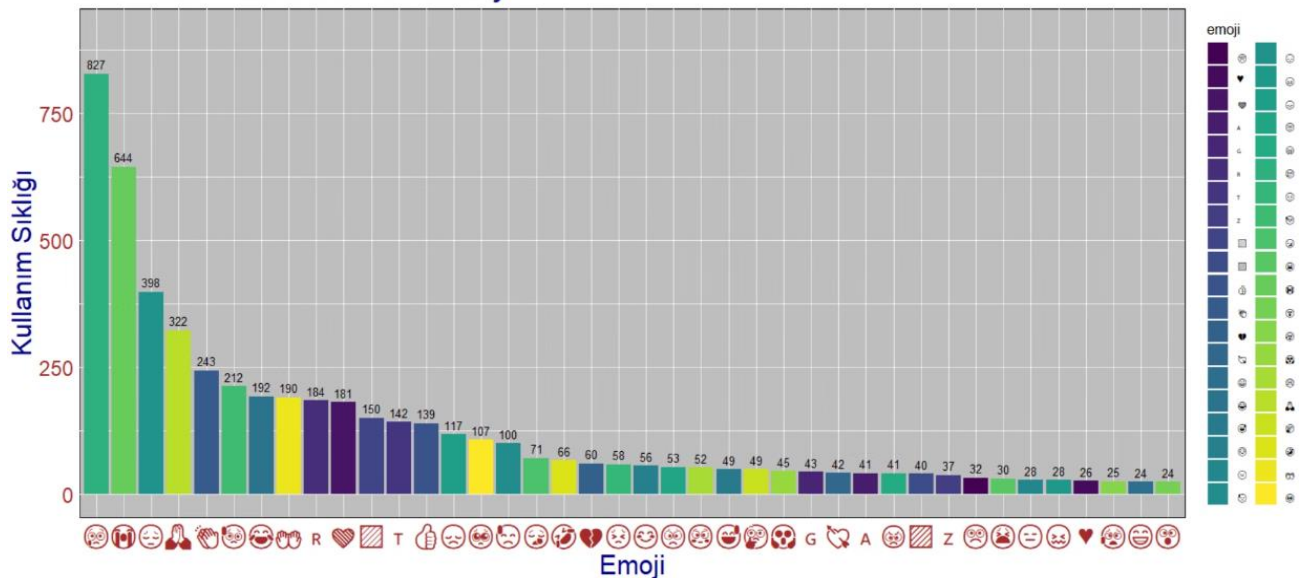
```
top_40_emojiler$emoji
```

Açıklama:

- top_40_emojiler veri çerçevesindeki emoji sütununu döndürerek, en yüksek frekansa sahip ilk 40 emojinin Unicode karakterlerini içeren bir vektör elde edilir

Çubuk Grafik

```
{r}
# Grafik oluşturma
ggplot(data = top_40_emojiler, aes(x = factor(emoji, levels = emoji), y = frequency, fill = emoji)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = frequency), vjust = -0.5, color = "black", size = 3, angle = 0, position = position_dodge(width = 1)) + # Metinleri yatay olarak bırak
labs(title = "En Fazla Kullanılan 40 Emoji", x = "Emoji", y = "Kullanım Sıklığı") +
scale_x_discrete(labels = top_40_emojiler$emoji) +
theme_minimal() +
  theme(panel.background = element_rect(fill = "grey"),
        axis.text.x = element_text(size = 20, hjust = 0.5, color = "brown"), # Hizalama düzeltildi
        axis.text.y = element_text(size = 15, color = "brown"),
        plot.title = element_text(color = "darkblue", size = 25),
        axis.title.x = element_text(color = "darkblue", size = 20),
        axis.title.y = element_text(color = "darkblue", size = 20)) +
coord_cartesian(ylim = c(0, max(top_40_emojiler$frequency) * 1.1)) +
scale_fill_viridis_d() # Renkli sütunlar
```



Emoji Bulutu

```
## Emoji bulutu oluşturalım.
library(tidyverse)

emoji_data <- data.frame(word = emoji_labels, freq = emoji_counts)

wordcloud2(data = emoji_data, size = 3, color = "random-dark", backgroundColor = "white", shape = "circle", fontWeight = "bold", minSize = 5)
```

```
emoji_data <- data.frame(word = emoji_labels, freq = emoji_counts)
wordcloud2(data = emoji_data, size = 3, color = "random-dark", backgroundColor =
"white", shape = "circle", fontWeight = "bold", minSize = 5)
```

Açıklama:

- wordcloud2() fonksiyonu kullanılarak emoji_data veri çerçevesi üzerinde bir emoji bulutu oluşturulur.
- size, color, backgroundColor, shape, fontWeight, ve minSize gibi parametrelerle bulutun görünümü ayarlanır.



Duygu Analizi

```
# tidytext paketini kullanarak Bing Liu'nun duygu analizine yönelik sözlüğünü getirir.
library(tidytext)
duygu_sozlugu <- get_sentiments("bing")
```

```
duygu_sozlugu <- get_sentiments("bing")
```

Açıklama:

- get_sentiments() fonksiyonuyla tidytext paketinden Bing Liu'nun duygu analiz sözlüğü yüklenir.
-

```
#Bu kod bloğu, deprem adlı bir veri çerçevesindeki emojiilerin duygu analizini gerçekleştirir.
library(tidytext)
metin_uygu <- deprem %>%
  unnest_tokens(word, metin) %>%
  inner_join(duygu_sozlugu, by = "word")
```

```
metin_uygu <- deprem %>%
  unnest_tokens(word, metin) %>%
  inner_join(duygu_sozlugu, by = "word")
```

Açıklama:

- unnest_tokens() ve inner_join() fonksiyonları kullanılarak deprem veri çerçevesindeki metinlerdeki emojiilerin duygu analizi yapılır

```
library(tidytext)
duygu_sayilari <- metin_uygu %>%
  count(sentiment)
print(duygu_sayilari)
```

A tibble: 2 × 2

| sentiment
<chr> | n
<int> |
|--------------------|------------|
| negative | 334 |
| positive | 210 |

2 rows

```
duygu_sayilari <- metin_uygu %>%
  count(sentiment)
print(duygu_sayilari)
```

Açıklama:

- count() fonksiyonu ile metin_uygu veri çerçevesindeki duygu türlerinin sayıları hesaplanır.


```
#Bu kod bloğu, duygu_sayilari tablosundaki her bir duygu türünün yüzdesini hesaplar ve bu bilgiyi tabloya ekler.
```{r}
toplam_duygu_sayisi <- sum(duygu_sayilari$n)
duygu_sayilari <- duygu_sayilari %>%
 mutate(yuzde = (n / toplam_duygu_sayisi) * 100)
print(duygu_sayilari)
```
```

A tibble: 2 × 3

| sentiment
<chr> | n
<int> | yuzde
<dbl> |
|--------------------|------------|----------------|
| negative | 334 | 61.39706 |
| positive | 210 | 38.60294 |

2 rows

```
toplam_duygu_sayisi <- sum(duygu_sayilari$n)
duygu_sayilari <- duygu_sayilari %>%
  mutate(yuzde = (n / toplam_duygu_sayisi) * 100)
print(duygu_sayilari)
```

Açıklama:

- Duygu türlerinin yüzdelik dağılımını hesaplamak için mutate() ve sum() fonksiyonları kullanılır.

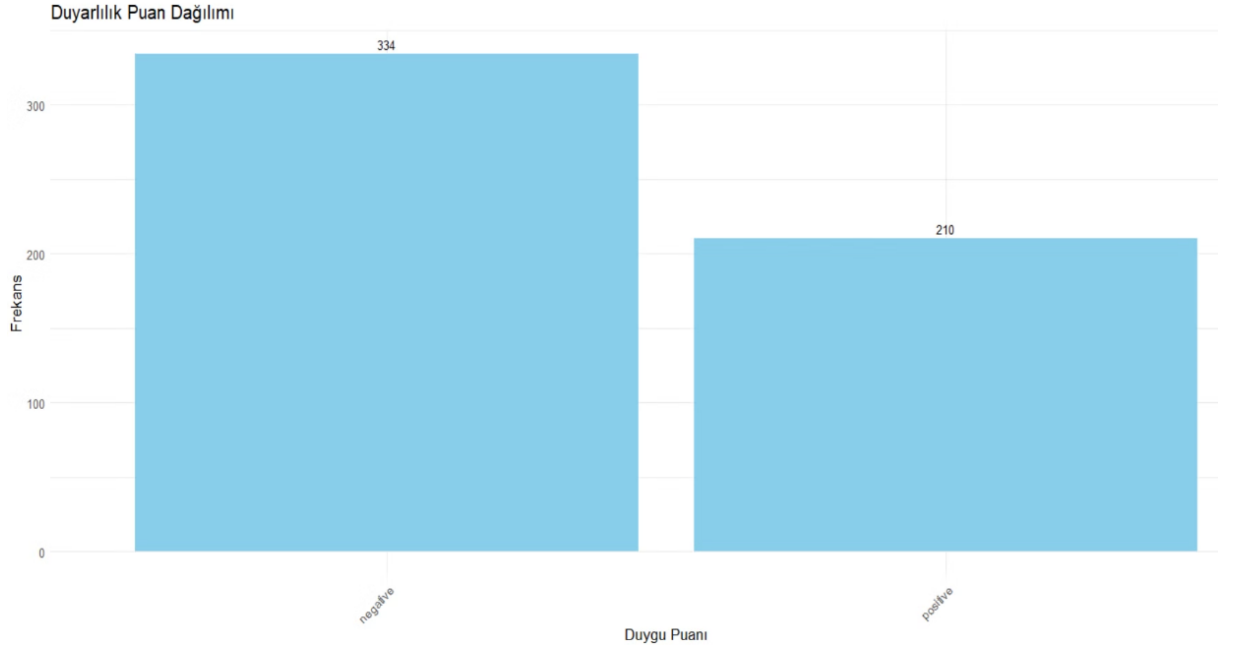
Duygu Analizinin Görselleştirilmesi

```
#Bu kod bloğu, duygu_sayilari verisini kullanarak duygu puanlarının dağılımını görselleştiren bir çubuk grafik oluşturur.
```{r}
ggplot(duygu_sayilari, aes(x = sentiment, y = n)) +
 geom_bar(stat = "identity", fill = "skyblue") +
 geom_text(aes(label = n), vjust = -0.5, color = "black", size = 3) + # Sütunların içine frekansları ekler
 labs(title = "Duyarlılık Puan Dağılımı", x = "Duygu Puanı", y = "Frekans") +
 theme_minimal() +
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
```

```
ggplot(duygu_sayilari, aes(x = sentiment, y = n)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = n), vjust = -0.5, color = "black", size = 3) + # Sütunların içine
  frekansları ekler
  labs(title = "Duyarlılık Puan Dağılımı", x = "Duygu Puanı", y = "Frekans") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Açıklama:

- ggplot() fonksiyonu ile duygu puanlarının dağılımını görselleştirmek için bir çubuk grafik oluşturulur.
- geom_bar() ile çubuklar oluşturulur ve geom_text() ile her bir çubuğun üzerine frekans değerleri eklenir.
- labs() ile grafik başlığı ve eksen etiketleri belirlenir.
- theme_minimal() ile minimal bir tema seçilir ve axis.text.x ile x eksenindeki etiketlerin açısını ve hizasını düzenleriz.



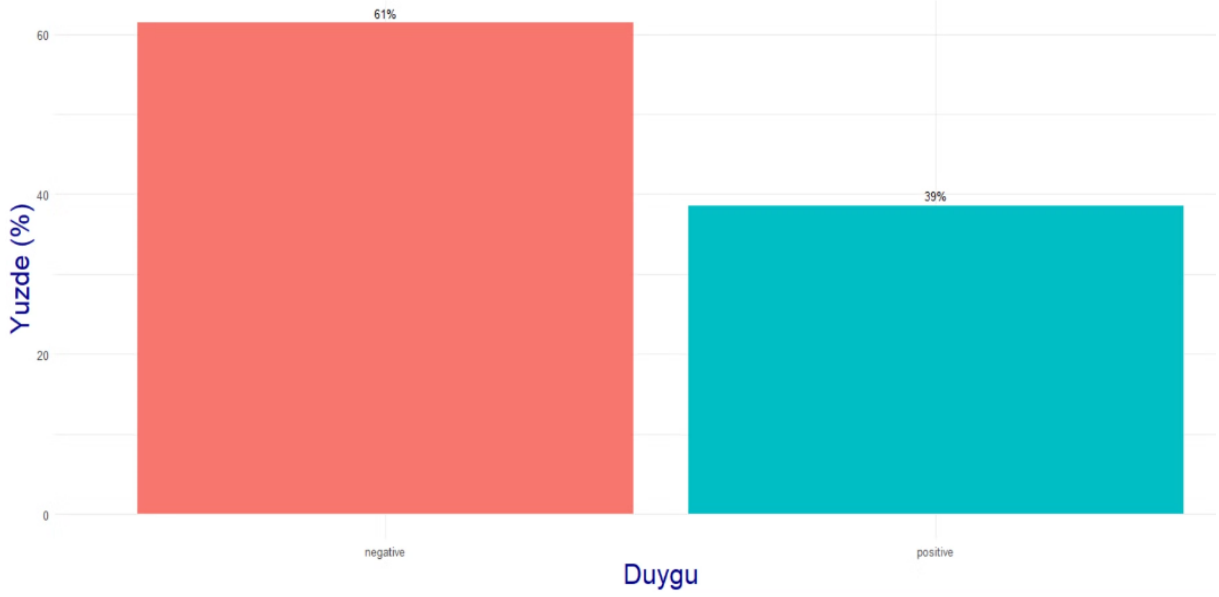
#Bu kod bloğu, `duygu_sayilari` verisini kullanarak her bir duygu türünün yüzdesini gösteren bir çubuk grafik oluşturur.

```
...[r]
ggplot(data = duygu_sayilari, aes(x = sentiment, y = yuzde, fill = sentiment)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(yuzde), "%")), vjust = -0.5, color = "black", size = 3) + # Sutunlarin icine yuzdeleri ekler
  labs(title = "", x = "Duygu", y = "Yuzde (%)") +
  theme_minimal() +
  theme(plot.title = element_text(color = "darkblue", size = 25),
        axis.title.x = element_text(color = "darkblue", size = 20),
        axis.title.y = element_text(color = "darkblue", size = 20),
        legend.position = "none")
...
```

```
ggplot(data = duygu_sayilari, aes(x = sentiment, y = yuzde, fill = sentiment)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = paste0(round(yuzde), "%")), vjust = -0.5, color = "black", size
= 3) + # Sutunlarin icine yuzdeleri ekler
  labs(title = "", x = "Duygu", y = "Yuzde (%)") +
  theme_minimal() +
  theme(plot.title = element_text(color = "darkblue", size = 25),
        axis.title.x = element_text(color = "darkblue", size = 20),
        axis.title.y = element_text(color = "darkblue", size = 20),
        legend.position = "none")
```

Açıklama:

- İkinci bir çubuk grafik oluşturulur, bu grafikte `yuzde` değişkeni kullanılarak her duygu türünün yüzdesi gösterilir.
- `ggplot()` ile grafiği oluşturulur, `geom_bar()` ile çubuklar çizilir ve `geom_text()` ile `yuzde` değerleri her çubuğun üzerine eklenir.
- `labs()` ile grafik başlığı ve eksik etiketleri belirlenir.
- `theme_minimal()` ile minimal bir tema seçilir ve `plot.title`, `axis.title.x`, `axis.title.y`, ve `legend.position` ile temalar düzenlenir.



Emojilerin Anlamları ve Kullanım Sebepleri

1. 😞 (Ağlayan yüz):

Sebeb: Depremin sebep olduğu kayıplar ve yıkım nedeniyle derin üzüntü ve acıyı ifade eder. Bu emoji, insanların duygusal tepkilerini ve yaslarını dile getirmek için kullanılır.

2. 😭 (Ağlayan yüz, gözlerinden yaş gelen):

Sebeb: Büyük bir üzüntü ve acı ifade eder. İnsanlar, depremde yaşanan trajediler karşısında duygularını daha yoğun bir şekilde ifade etmek için bu emojiyi kullanır.

3. 😔 (Düşünen yüz):

Sebeb: Keder ve hüznü ifade eder. Deprem sonrası yaşanan travma ve yas sürecinde, insanların içsel düşüncelerini ve duygusal durumlarını yansıtır.

4. 🙏 (Dua eden eller):

Sebeb: Depremde etkilenenler için dua etmek ve destek göstermek amacıyla kullanılır. Bu emoji, insanların dayanışma ve umut dileklerini ifade eder.

5. 🙌 (Alkış):

SebeP: Kurtarma ekiplerine, gönüllülere ve dayanışma gösteren kişilere destek ve teşekkür amacıyla kullanılır. Bu emoji, yapılan çalışmaları takdir etmek için kullanılır.

6. 😞 (Üzgün yüz):

SebeP: Üzüntü ve kederi ifade eder. Depremde yaşanan kayıplar ve trajediler karşısında insanların hissettiğı duyguları yansıtır.

7. 😏 (Gülen yüz, gözlerinden yaş gelen):

SebeP: Trajikomik durumlar veya şok karşısında ironiyle tepki vermek için kullanılır. Bazen insanlar, trajediler karşısında gülme refleksi gösterebilir.

8. 🙏 (Açık eller, dua):

SebeP: Yardım istemek veya dua etmek amacıyla kullanılır. Bu emoji, insanların umudunu ve yardım dileklerini ifade eder.

9. 🇹🇷 (Bayrak):

SebeP: Milliyetçilik ve birlik duygularını ifade eder. Depremden etkilenen ülkeye destek göstermek amacıyla kullanılır.

10. ❤️ (Kırmızı kalp):

SebeP: Sevgi ve destek göstermek amacıyla kullanılır. Depremden etkilenenlere karşı duyulan sevgiyi ifade eder.

Sonuç

Marmara Depremi, Türkiye'nin tarihinde en büyük doğal felaketlerden biri olarak kaydedildi. 17 Ağustos 1999'da gerçekleşen bu deprem, binlerce can kaybına, yüzbinlerce kişinin evsiz kalmasına ve büyük bir maddi hasara yol açtı. Bu tür büyük felaketler, toplumun duygusal ve psikolojik tepkilerini çeşitli şekillerde ifade etmelerine neden olur. Sosyal medya platformları, insanların bu duygusal ifadelerini paylaşmak için sıkça başvurdukları yerler haline gelmiştir ve burada kullanılan emoji, duygusal ifadelerin önemli bir bileşeni olarak öne çıkar.

Depremi ardından, insanlar sosyal medya aracılığıyla duygularını ifade ederken çeşitli emoji kullanır. Bu emoji, bir dizi duyguyu, düşünceyi ve durumu temsil eder. Özellikle kriz ve felaket dönemlerinde, kullanılan emoji insanların içinde bulundukları ruh halini yansıtmakta önemli bir araçtır. Bazı emoji olumlu duyguları, umudu ve dayanışmayı yansıtırken, diğeri ise üzüntü, korku ve öfke gibi negatif duyguları ifade eder. Bu emoji, toplumun deprem gibi büyük felaketlere nasıl tepki verdiğini ve bu süreçte nasıl bir araya geldiğini anlamamıza yardımcı olur.

Pozitif duyguları ifade eden emoji arasında 🙌 dua eden eller, 🙏 dua eden insan, 🙌 alkış ve ❤️ kırmızı kalp bulunur. Bu emoji, insanların destek, dua ve sevgi göstermelerine olanak tanır. Özellikle dua eden eller ve dua eden insan emoji, depremzedelere yönelik iyi dilekleri ve manevi desteği ifade etmek için sıkça kullanılır. Alkış emoji, kurtarma ekiplerinin ve gönüllülerin çabalarını takdir etmek için yaygın olarak kullanılırken, kırmızı kalp emoji sevgi ve dayanışmayı sembolize eder.

Negatif duyguları yansıtan emoji arasında ise 😡 kızgın yüz, 😞 hüzünlü yüz ve ☹ somurtan yüz yer alır. Bu emoji, öfke, hüznü ve memnuniyetsizlik gibi duyguları ifade eder ve toplumun yaşadığı acıyı ve zorluğu belirtir. Kızgın yüz emoji, genellikle felaketin ardından yetkililere veya altyapı eksikliklerine yönelik öfkeyi ifade etmek için kullanılır. Hüzünlü ve somurtan yüz emoji ise kaybedilen hayatlar ve yaşanan travmalar karşısında duyulan derin üzüntüyü ve kederi yansıtır.

Ayrıca, bayrak emoji de (R-T-G-A-Z) sıkça kullanılır. Özellikle 🇹🇷 (Türkiye), 🇬🇷 (Yunanistan), 🇿🇦 (Güney Afrika) ve 🇦🇿 (Azerbaycan) gibi bayrak emoji, depremzedelere destek göstermek ve milli birlik duygularını ifade etmek amacıyla kullanılır. Bu emoji, ulusal ve uluslararası destek ve dayanışmayı vurgulamak için önemli bir araçtır. Bayrak emoji, toplumun birlik ve beraberlik içinde olduğunu gösterir ve milli duyguları pekiştirir.

Sosyal medya platformlarında kullanılan emojiiler, iletiřimde duygusal derinlięi artırır ve insanların birbirlerine destek olmalarını saęlar. Bu emojiiler, toplumun farklı kesimlerinden insanların, deprem gibi büyük felaketler karřısında nasıl bir araya geldięini ve birbirlerine nasıl destek olduklarını gösterir. Ayrıca, bu emojiiler, insanların duygusal durumlarını daha açık ve etkili bir şekilde ifade etmelerine yardımcı olur.

Verilen emojiilerin yüzde dağılımını incelediğimizde, Marmara Depremi ile iliřkilendirilen yorumlarda çoęunlukla negatif duyguların ön planda olduęunu görebiliriz. Yorumlarda üzüntü, keder ve endiře gibi negatif duyguları ifade eden emojiiler aęırlıklı olarak kullanılmıřtır. Bu, toplumun yařadığı büyük kayıplar ve travmaların bir yansımasıdır. Negatif duyguları ifade eden emojiiler 334 kez kullanılmıřtır, bu da genel yorumların %61'ini oluřturmaktadır.

Bununla birlikte, umut, destek ve dayanıřma gibi pozitif duyguları yansıtan emojiiler de önemli bir yer tutmaktadır. Pozitif duyguları ifade eden emojiiler 210 kez kullanılmıřtır ve bu da yorumların %39'unu oluřturmaktadır. Bu sonuçlar, Marmara Depremi'ne iliřkin sosyal medya yorumlarının genel olarak negatif duygularla yoęunlařtıęını, ancak pozitif duyguların da önemli bir şekilde varlıęını sürdürdüęünü göstermektedir. Bu durum, felaket zamanlarında toplumun karmařık ve çeřitli duygusal tepkilerle bařa çıktıęını ve birlikte dayanıřma içinde olduęunu vurgular.

Sonuç olarak, Marmara Depremi gibi büyük felaketler, toplumun çeřitli duygusal tepkiler göstermesine neden olur ve bu duyguların ifade edilmesinde sosyal medya ve emojiiler önemli bir rol oynar. Emojiiler, duygusal ifadeleri güçlendiren ve insanların birbirlerine destek olmalarını saęlayan araçlardır. Depremın ardından kullanılan emojiiler, toplumun yařadığı acıyı, umudu ve dayanıřmayı yansıtarak, duygusal bağları güçlendirir ve toplumsal birlik ve dayanıřmanın altını çizerek. Bayrak emojiileri, ulusal ve uluslararası desteęi ve dayanıřmayı vurgulamak için önemli bir sembol olarak kullanılır. Bu analiz, sosyal medya platformlarında kullanılan emojiilerin, kriz zamanlarında toplumun duygusal tepkilerini ve ruh halini anlamada ne kadar önemli olduęunu bir kez daha gözler önüne serer.

Kaynakça

<https://www.youtube.com/watch?v=17Uj9eLHCDY>

HER 250 SENEDE YAŞANAN BÜYÜK MARMARA DEPREMİ! (SON DEPREM 257 YIL ÖNCEYDİ)

<https://www.youtube.com/watch?v=jntEjPEFgmg>

Marmara depremi: İstanbul'u neler bekliyor? Uzmanlar neler söylüyor? Hazırlıklar ne durumda?

<https://www.youtube.com/watch?v=uUakx0hRFGI>

Büyük Deprem Nerede, Ne Zaman ve Ne Büyüklükte Olacak? | Deprem Tahmini!

<https://www.youtube.com/watch?v=hcP5b377tpQ>

Beklenen Büyük Marmara Depreminin simülasyonu bize ne anlatıyor? En çok nereler etkilenecek?

<https://www.youtube.com/watch?v=qHDORKGMakk>

Naci Görür'den Korkutan Marmara Depremi Uyarısı! Etkilenecek İlleri Tek Tek Açıkladı

<https://www.youtube.com/watch?v=XhSt2ZfO-5A>

Naci Görür'den Marmara Depremi İçin Önemli Uyardı! | NTV

<https://www.youtube.com/watch?v=DUMC2CbFfhA>

Marmara'da hangi fay hatları aktif? Prof. Dr. Hüseyin Öztürk değerlendirdi

<https://www.youtube.com/watch?v=XI09KW481M8>

17 Ağustos Depremi (İLK 5 DAKİKASI) Canlı Yaşananlar

<https://www.youtube.com/watch?v=SYzs1TaxRxY>

Deprem: O an - 17 Ağustos 1999 Depremi'ni yaşayanlar anlatıyor

<https://www.youtube.com/watch?v=hWuYE7LLUQE>

17 Ağustos 1999 Marmara depremi | Büyüteç

<https://www.youtube.com/watch?v=R4FGV-f1xfs>

17 ağustos 1999

https://www.youtube.com/watch?v=ea_OKowB1Vc

17 Ağustos Depremi - Gölcük | TRT Arşiv

<https://www.youtube.com/watch?v=30a9qKc3ZLw>

17 Ağustos 1999 depreminde yıkılmayan yer Tavşancıl

https://www.youtube.com/watch?v=LjimpRF_XiP8

Gölcük Depremi | 17 Ağustos 1999 | 32.Gün Arşivi

<https://www.youtube.com/watch?v=449wwLDxwdg>

17 Ağustos 1999 Gecesi Neler Yaşandı? | 32. Gün Arşivi

<https://www.youtube.com/watch?v=LTo6X6Eek8M>

45 saniye: 17 ağustos 1999 marmara depremi

<https://www.youtube.com/watch?v=mSfVkvafIp0>

Deprem: 17 Ağustos Depremi'nin 20. yılında Gölcük-Değirmendere'de yaşam

https://www.youtube.com/watch?v=BQnQNOURG_c

17 Ağustos Depreminin Yeni Telsiz Kayıtları Ortaya Çıktı

<https://www.youtube.com/watch?v=C6nxJNZAQNI>

17 Ağustos Depreminde İnanılmaz Kurtuluşun O Anı

rpubs.com/nur_kuban/emoji

Nur Kuban Torun

<https://chatgpt.com>