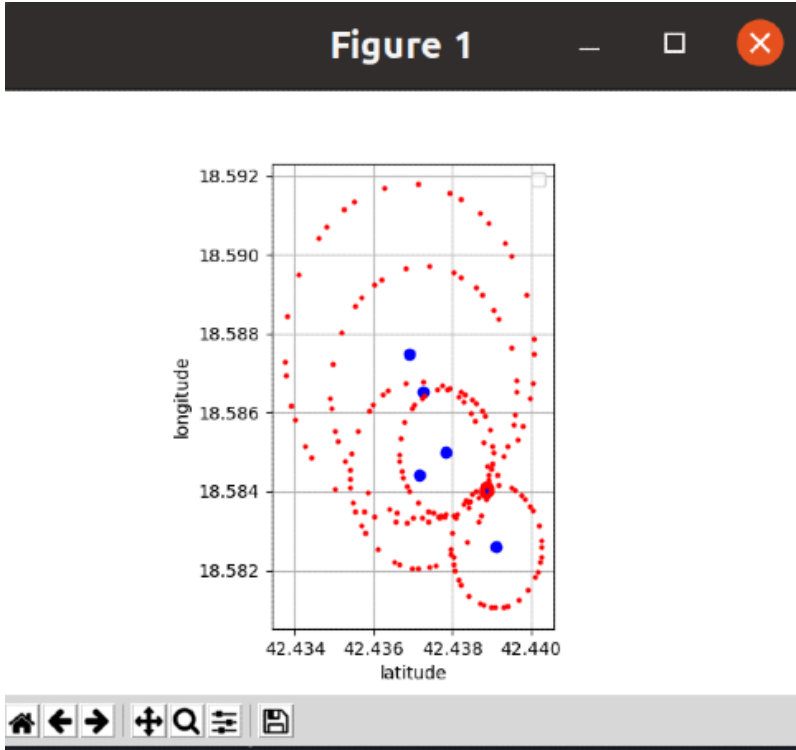


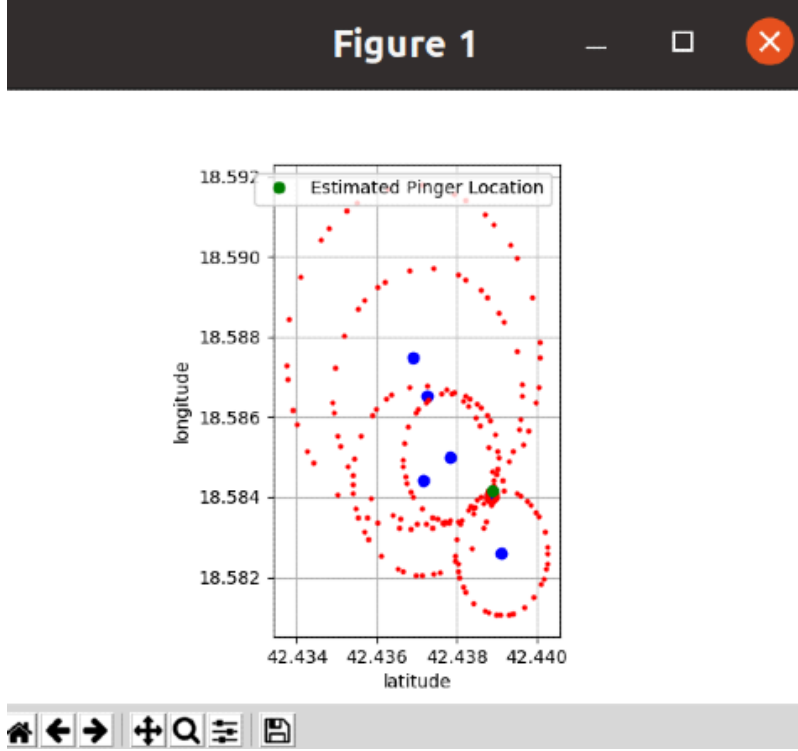
# ITU AUV Ödev Raporu 3 - Talha Karasu

Ödevde `data.csv` dosyasındaki verileri inceleyerek başladım. `lat`, `lon`, `depth` ve `dist` değerlerini nasıl kullanabileceğimi düşündüm. İlk aşamada matplotlib ile elimizdeki veriyi görselleştirmeye başladım. Dataframe'e kaydettiğim konum verilerini grafik üzerinde nokta olarak gösterdim. Bu aşamada ödevdeki örnek haritada olduğu gibi her noktanın etrafına aldığımız uzaklık verisini yarıçap kabul eden bir daire çizmeye çalıştım. `lat` ve `lon` değerlerini bildiğimiz iki nokta arasındaki mesafeyi Haversine formülü ile hesaplayabildiğimi öğrendim. Uzaklık ve bir noktanın konumunu bildiğimiz için bu formülü düzenleyerek 3. değişkeni bulabileceğimizi fark ettim. Python'daki `haversine` paketinin `inverse_haversine` fonksiyonu ile konum, uzaklık ve açı değerlerini kullanarak diğer noktanın konumunu hesapladığımı gördüm. Bir loop ile 0 360 arasında 10 step değeriyle ilerledim, o açıdaki noktayı `inverse_haversine` fonksiyonuyla hesapladım, bu sayede her noktanın etrafında kesikli bir daire oluşturdum. Bu aşamada bazı dairelerin bir birleriyle kesişmediğini fark ettim, verilerdeki distance değerlerinin hatalı olduğunu ve 2 katlarını almam gerektiğini öğrendikten sonra bu aşamayı tamamladım.

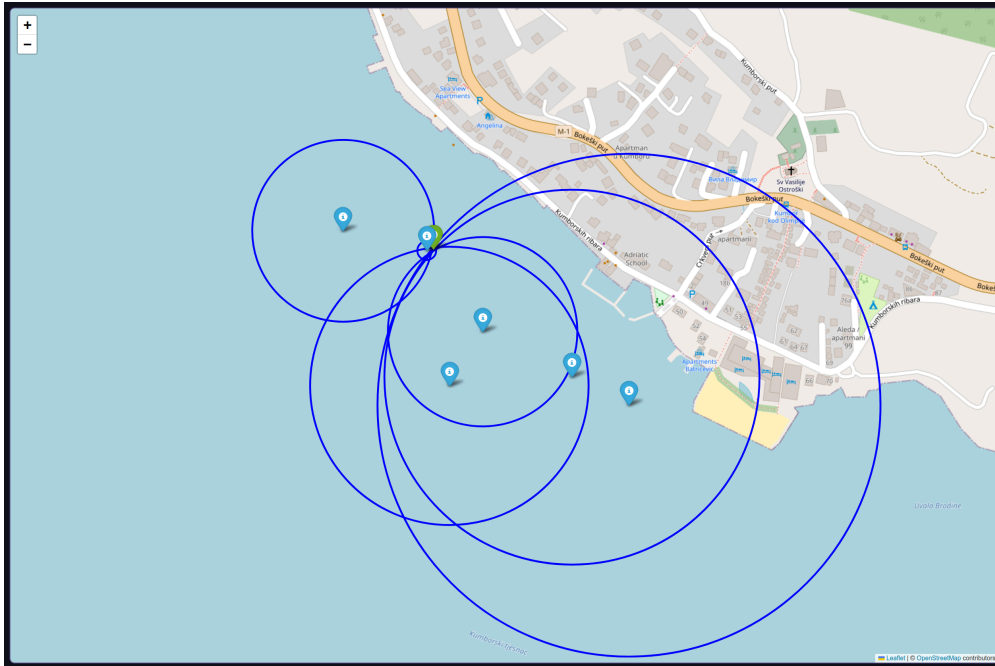


Least squares kullanarak kürelerin kesiştikleri noktayı hesaplamak için pythondaki `scipy` paketinin `least_squares` fonksiyonunu kullandım. Bu fonksiyon bizden hataları hesaplamak için bir fonksiyon istediği için `error_function` fonksiyonunu yazdım. Bu fonksiyon parametre olarak bulmak istediğimiz konumu (pinger), bildiğimiz noktaları ve uzaklıkları alıyor. Her noktanın pinger'a olan yeryüzü uzaklığını Haversine formülü kullanarak hesaplıyor, pisagor ile derinlik ve önceki adımda hesaplanan

uzaklığı kullanarak iki nokta arasındaki toplam uzaklığı hesaplıyor. Bu hesaplanan toplam uzaklıktan en baştaki veride bilinen uzaklığı çıkararak hatayı hesaplıyor. `least_squares` fonksiyonu bu hataları kullanarak pingerin olma ihtimali en yüksek olan konumu buluyor. Bu konumu da matplotlib üzerinde doğruladıktan sonra bu aşamayı da tamamladım.



Sonraki aşamada `folium` paketini kullanarak bu verileri gerçek bir harita üzerinde görselleştirdim. Oluşturduğum haritayı `map.html` dosyasına kaydederek ödevi tamamladım.



[Proje dosyalarına bu repodan ulaşabilirsiniz.](#)