**1. [1.] Resolving inconsistencies [Tutarsızlıkları çözme]**

In the previous exercise, you identified a dip in conversion rates for House Ads. [Önceki alıştırmada, Kurum Reklamları için dönüşüm oranlarında bir düşüş tespit etmiştiniz.] It appears that the problem was that users were seeing ads in languages other than there preferred language. [Sorunun, kullanıcıların tercih ettikleri dilden farklı dillerde reklam görmeleri olduğu anlaşılıyor.] In this lesson, we'll assess the impact of this mistake. [Bu derste, bu hatanın etkisini değerlendireceğiz.]

**2. [2.] Assessing impact [Etkiyi değerlendirmek]**

While you cannot ignore data related to errors in the campaign, you can estimate what conversion might have looked like if there had been no issues. [Kampanyadaki hatalarla ilgili verileri görmezden gelemezsiniz, ancak herhangi bir sorun olmasaydı dönüşümün nasıl görüneceğini tahmin edebilirsiniz.] One way to assess impact is to index all other languages' conversion rates to English during the period where the ads were running in the correct language for each user. [Etkiyi değerlendirmenin bir yolu, reklamların her kullanıcı için doğru dilde yayınlandığı dönemde diğer tüm dillerin dönüşüm oranlarını İngilizce'ye endekslemektir.] We begin by slicing the house\_ads DataFrame to include the rows where the date\_served is prior to when the language bug arose. [Dil hatasının ortaya çıktığı zamandan önce date\_served olduğu satırları dahil etmek için house\_ads DataFrame'i dilimleyerek başlıyoruz.] Using our conversion\_rate() function, we calculate conversion rate for each language during that period. [convert\_rate() işlevimizi kullanarak, o dönem boyunca her dil için dönüşüm oranını hesaplarız.]

**3. [3.] Assessing impact [Etkiyi değerlendirmek]**

We then divide the conversion rate of all other languages by the conversion rate of English in order to understand the relative relationship of how well our marketing assets typically convert users for each language compared to English. [Daha sonra, pazarlama varlıklarımızın her dil için kullanıcıları İngilizce'ye kıyasla ne kadar iyi dönüştürdüğüne ilişkin göreceli ilişkiyi anlamak için diğer tüm dillerin dönüşüm oranını İngilizce'nin dönüşüm oranına böleriz.]

**4. [4.] Interpreting Indexes [İndeksleri Yorumlama]**

What these indexes mean is that Spanish-speaking users typically convert 1.7 times the rate of English-speakers and Arabic and German speakers convert at about 4-5 times the rate compared to English-speakers. [Bu endekslerin anlamı, İspanyolca konuşan kullanıcıların genellikle İngilizce konuşanlara göre 1,7 kat, Arapça ve Almanca konuşanlar ise İngilizce konuşanlara kıyasla yaklaşık 4-5 kat daha fazla dönüşüm sağladığıdır.]

**5. [5.] Daily conversion [Günlük dönüşüm]**

Next, we calculate the total number of users and actual conversions on each day. [Ardından, her gündeki toplam kullanıcı sayısını ve gerçek dönüşümleri hesaplarız.] First, we group the DataFrame by date\_served and language\_preferred. [İlk olarak, DataFrame'i date\_served ve language\_preferred'e göre gruplandırıyoruz.] Next, we do something different. [Ardından, farklı bir şey yapıyoruz.] We use the agg() method since we wish to calculate multiple statistics. [Birden fazla istatistik hesaplamak istediğimiz için agg() yöntemini kullanıyoruz.] We pass a dictionary to this method where the key is the column name, and the value is the method we want to apply on the column. [Anahtarın sütun adı olduğu ve değerin sütuna uygulamak istediğimiz yöntem olduğu bu yönteme bir sözlük geçiyoruz.] Thus, we calculate the total number of unique user ids and the total number of users who converted. [Böylece, benzersiz kullanıcı kimliklerinin toplam sayısını ve dönüşüm gerçekleştiren toplam kullanıcı sayısını hesaplıyoruz.]

**6. [6.] Daily conversion [Günlük dönüşüm]**

Finally, we unstack our result with level equals one to make it easier to manipulate in future steps. [Son olarak, ilerideki adımlarda manipüle etmeyi kolaylaştırmak için sonucumuzun seviye eşittir bir ile yığınını kaldırırız.] The result is a DataFrame with the number of users who should have seen ads in each language and how many of those users converted each day. [Sonuç, her dilde reklam görmesi gereken kullanıcı sayısını ve bu kullanıcıların kaçının her gün dönüşüm sağladığını gösteren bir DataFrame'dir.]

**7. [7.] Create English conversion rate column [İngilizce dönüşüm oranı sütunu oluşturun]**

Since the conversion\_rate() function puts the date\_served in the DataFrame's index, we can use the loc accessor to slice our DataFrame and retrieve columns only from the period where the language bug was a problem. [convert\_rate() işlevi data\_served öğesini DataFrame'in dizinine koyduğundan, loc erişimcisini DataFrame'imizi dilimlemek ve yalnızca dil hatasının sorun olduğu dönemden sütunları almak için kullanabiliriz.] Our DataFrame has multi-level column names, so we can access the total number of people who converted for the English language by putting the two names in parentheses as a set, first writing "converted" because that is the first level of the column structure and then the relevant language. [DataFrame'imiz çok seviyeli sütun adlarına sahiptir, bu nedenle, iki adı parantez içine alarak, önce 'dönüştürülmüş' yazarak İngilizce diline dönüştüren toplam kişi sayısına erişebiliriz, çünkü bu, sütun yapısının ilk seviyesidir. ve ardından ilgili dil.] In this case, converted comma English. [Bu durumda, dönüştürülen virgül İngilizce.]

**8. [8.] Calculating daily expected conversion rate [Günlük beklenen dönüşüm oranını hesaplama]**

Next, we can multiply the actual English conversion rate during this time by the language indexes we created earlier to determine what the expected conversion rates for these languages would have been for each day. [Ardından, bu diller için her gün için beklenen dönüşüm oranlarının ne olacağını belirlemek için bu süre boyunca gerçek İngilizce dönüşüm oranını daha önce oluşturduğumuz dil dizinleriyle çarpabiliriz.]

**9. [9.] Calculating daily expected conversions [Günlük beklenen dönüşümleri hesaplama]**

Then, we multiply the daily expected conversion rate of each language by the number of users who should have seen ads in that language. [Ardından, her dilin günlük beklenen dönüşüm oranını, o dilde reklam görmüş olması gereken kullanıcı sayısıyla çarpıyoruz.] This gives us how many subscribers we would have expected if the language bug had not occurred. [Bu bize, dil hatası oluşmamış olsaydı, beklediğimiz abone sayısını verir.]

**10. [10.] Determining the number of lost subscribers [Kayıp abone sayısının belirlenmesi]**

To calculate the overall impact, limit the expected conversion dataset to only the days when the bug occurred. [Genel etkiyi hesaplamak için, beklenen dönüşüm veri kümesini yalnızca hatanın oluştuğu günlerle sınırlayın.] Next, sum the number of expected and actual subscribers during that period, individually. [Ardından, o dönemde beklenen ve gerçekleşen abone sayısını ayrı ayrı toplayın.] Finally, we take the difference in the number of subscribers we expected and the subscribers we received, which gives us an estimate of how many subscribers we lost due to the language error. [Son olarak, beklediğimiz abone sayısı ile aldığımız abone sayısı arasındaki farkı alıyoruz ve bu da bize dil hatası nedeniyle kaç abone kaybettiğimizi tahmin ediyor.]

**11. [11.] Let's practice! [Hadi pratik yapalım!]**

Now it's your turn. [Şimdi senin sıran.] Let's assess the impact of this bug! [Bu hatanın etkisini değerlendirelim!]