**1. [1.] Building functions to automate analysis [Analizi otomatikleştirmek için işlevler oluşturma]**

In the previous chapter, you may have noticed that you were doing a lot of similar, repetitive tasks. [Önceki bölümde, birçok benzer, tekrarlayan görev yaptığınızı fark etmiş olabilirsiniz.] Anytime you notice repetition in your work, you should think about how you can automate it. [İşinizde tekrarı fark ettiğinizde, onu nasıl otomatikleştirebileceğinizi düşünmelisiniz.]

**2. [2.] Why build a function? [Neden bir işlev inşa etmelisiniz?]**

One of the best ways to make this kind of analysis faster is to create functions. [Bu tür analizleri daha hızlı hale getirmenin en iyi yollarından biri işlevler oluşturmaktır.] For example, remember when you analyzed the quality of subscribers by day and conversion channel in the last chapter? [Örneğin, geçen bölümde abonelerin kalitesini ve dönüşüm kanalını gün bazında analiz ettiğinizi hatırlıyor musunuz?] You might want to conduct this kind of analysis repeatedly for different subsegments of the customer base. [Müşteri tabanının farklı alt bölümleri için bu tür analizleri tekrar tekrar yapmak isteyebilirsiniz.]

**3. [3.] Print daily\_retention\_rate [Daily\_retention\_rate yazdır]**

Rather than copy-paste the code snippet to make the appropriate edits, which can lead to typos and make it difficult to correct bugs as they arise, it is better to write a function. [Yazım hatalarına yol açabilecek ve ortaya çıkan hataları düzeltmeyi zorlaştırabilecek uygun düzenlemeleri yapmak için kod parçacığını kopyalayıp yapıştırmak yerine, bir işlev yazmak daha iyidir.]

**4. [4.] Building a retention function [Tutma işlevi oluşturma]**

So we define a function, retention\_rate() that allows users to input a DataFrame and a list of column names. [Bu nedenle, kullanıcıların bir DataFrame ve bir sütun adları listesi girmesine izin veren bir işlev, retain\_rate() tanımlarız.] This function follows the same steps to calculate the total number of users who converted and retained. [Bu işlev, dönüşüm gerçekleştiren ve elde kalan kullanıcıların toplam sayısını hesaplamak için aynı adımları izler.] First, it calculates the total number of retained users, then the total number of subscribers, and finally divides them to obtain the retention rate. [İlk olarak, elde tutulan toplam kullanıcı sayısını, ardından toplam abone sayısını hesaplar ve son olarak, elde tutma oranını elde etmek için bunları böler.] However, this time in the groupby() method, we include the user-inputted column names. [Ancak bu sefer groupby() yönteminde kullanıcı tarafından girilen sütun adlarını dahil ediyoruz.]

**5. [5.] Retention rate by channel [Kanala göre tutma oranı]**

Now that we’ve defined the function, all we need to do to reproduce the retention rates from the previous chapter is call retention\_rate() with the marketing DataFrame, and pass the date\_subscribed and subscribing\_channel columns as a list of strings. [Artık işlevi tanımladığımıza göre, önceki bölümdeki elde tutma oranlarını yeniden oluşturmak için tek yapmamız gereken pazarlama DataFrame'i ile tutma\_rate() işlevini çağırmak ve bir dizi listesi olarak date\_subscribed ve subscribe\_channel sütunlarını geçirmektir.] After unstacking, we have the same results as before. [Yığını kaldırdıktan sonra, öncekiyle aynı sonuçlara sahibiz.]

**6. [6.] Plotting daily retention by channel [Kanala göre günlük tutma grafiğini çizme]**

Next, we follow the same steps as before to plot our results and... [Ardından, sonuçlarımızı çizmek için önceki adımların aynısını izleriz ve...]

**7. [7.] Messy daily retention rate chart [Dağınık günlük tutma oranı tablosu]**

Here's the resulting plot with much less effort than in the previous chapter due to our function! [İşlevimiz nedeniyle önceki bölümde olduğundan çok daha az çabayla ortaya çıkan arsa!] However, as you can see, this is a crowded chart that is nearly impossible to read. [Ancak, gördüğünüz gibi, bu, okunması neredeyse imkansız olan kalabalık bir tablodur.] I recommend looking at channels one-by-one to identify trends. [Trendleri belirlemek için kanallara tek tek bakmanızı öneririm.]

**8. [8.] Plotting function [çizim fonksiyonu]**

Again, instead of writing similar code over and over to plot the columns one at a time, we will create a function: this function will create several plots, one for each column. [Yine, sütunları birer birer çizmek için benzer kodu tekrar tekrar yazmak yerine, bir fonksiyon yaratacağız: bu fonksiyon, her sütun için bir tane olmak üzere birkaç çizim yaratacaktır.] The function uses a for loop to go through each column in the DataFrame and plot each column individually. [İşlev, DataFrame'deki her sütundan geçmek ve her sütunu ayrı ayrı çizmek için bir for döngüsü kullanır.] Note that here we are using matplotlib's plot() function to keep things simple. [Burada işleri basit tutmak için matplotlib'in plot() işlevini kullandığımızı unutmayın.] Since the dates are still located in the DataFrame's index, we use the index attribute to display dates on the x-axis, and the column values go on the y-axis. [Tarihler hala DataFrame dizininde bulunduğundan, tarihleri ​​x ekseninde görüntülemek için index niteliğini kullanırız ve sütun değerleri y ekseninde gider.] We can now call this function on the same daily\_channel\_retention DataFrame. [Artık bu işlevi aynı Daily\_channel\_retention DataFrame'de çağırabiliriz.]

**9. [9.] Email plot [E-posta planı]**

Which will then display a plot for each column in the DataFrame -- in this case a plot for each channel by date. [Bu daha sonra DataFrame'deki her sütun için bir çizim gösterecektir - bu durumda tarihe göre her kanal için bir çizim.] This is the plot for email. [Bu e-posta için arsa.] You can see email has big spikes that often go down to 0. [E-postanın genellikle 0'a kadar inen büyük artışları olduğunu görebilirsiniz.] This is common because emails are typically sent in bulk leading users to subscribe on the same set of limited days. [Bu yaygın bir durumdur, çünkü e-postalar genellikle toplu olarak gönderilir, önde gelen kullanıcılar aynı sınırlı günlerde abone olurlar.] When retention rate is 0, this means no one subscribed on those days. [Tutma oranı 0 olduğunda bu, o günlerde hiç kimsenin abone olmadığı anlamına gelir.]

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

Now it's your turn to practice building your own functions. [Şimdi kendi işlevlerinizi oluşturma alıştırması yapma sırası sizde.]