**1. Working with time series data in pandas**

Great job working through the mechanics of calculating KPIs. Those techniques will continue to be useful throughout the course.

**2. Exploratory Data Analysis**

In this chapter, we will discuss exploratory data analysis and working with time series data to uncover trends in KPIs.

**3. Review: Manipulating dates & times**

To start, let’s review some of the time and date manipulation techniques that we have briefly seen so far. Each of these has been shown but not fully explained.

**4. Example: Week Two Conversion Rate**

We will walk through an example of calculating the week two conversion rate, that is the rate at which people who have yet to subscribe, subscribe in their second week post lapse of the free trial.

**5. Using the Timedelta class**

To start, we must exclude users who have not yet been on the platform for two weeks. We do this by making sure the lapse date is less than our current date minus two weeks. To add or subtract an arbitrary interval to a date, in this case two weeks, we use the `timedelta` class. We create this by specifying a unit of time and a number of that unit. In this case we specify 14 days as above.

**6. Date differences**

Next, we find the number of days between a user’s lapsed and subscribed dates. We can simply subtract the two dates from one another to find this as we do above. This returns the difference between those two values in days. Here we have added this difference as the column `sub\_time`.

**7. Date components**

To convert this value to an integer rather than the unit of days we can extract the number with `dot-dt-dot-days`. We can also extract many other intervals, such as weeks or months. This is useful in a variety of ways as we will see through the remainder of this chapter.

**8. Conversion rate calculation**

Here let’s finish our conversion rate calculation. First we find the number of users who have not subscribed in week one and who have been on the platform two or more weeks. Then we find the number of those remaining users who have a `sub\_day` between 8 and 14. Finally, we can calculate our rate. As we can see this is close to 1%; not very high compared to the week one rate. As we will see, all of these techniques are very useful when working with the evolution of KPIs over time, or time based KPIs generally.

**9. Parsing dates - on import**

As a final note, dates can be represented in a variety of different string formats. The `read\_csv()` method has many options for parsing them automatically to the proper date type, that are worth looking into, and which we have been using in the background throughout this course. Two of the primary arguments, `parse\_dates` and `infer\_datetime\_format` are shown in use above. By setting these to true, `read\_csv()` will attempt to convert the string representation to a date on import.

**10. Parsing dates - manually**

Additionally, you can parse dates directly with the `to\_datetime` function. You can pass in the strings of your dates along with the strftime format representing that string in terms of date-components and separators. Above are some of the most common formats and their equivalent formatter string. This is worth understanding, as throughout the course of an analysis you will work with messy dates in a wide array of formats and it is good to know how to parse them to datetime objects correctly.

**11. Let's practice!**

Great work, now let’s practice mastering these techniques, as we will be applying them heavily in the remainder of this chapter.