Module 10
Video Transcripts

Video 1: The Data Cleaning Process (2:16)

Before you do any analysis on your data, you need to take steps to prepare it. This process is called data cleaning or data cleanse. More often than not you'll receive a dataset that's messy, has errors in it or has missing, duplicate or inconsistent data. And you have to take care of that first before you can analyze it. As a data scientist you'll often spend a lot more time cleaning data than running analyses on it. Let's discuss some examples of things you may need to clean in your data. Was this data simply not entered or does it not exist for that observation? Perhaps a blank actually represents the integer zero. Based on the requirements of your analysis, you may need to remove an observation from your dataset, if it has a missing data point. Or you may need to fill in that missing data point with an average or a zero. Second, your dataset could have duplicate data. Again, this could be due to a data entry mistake or due to data being updated. And resulting in multiple entries for the same observation. You need to determine which information's correct and remove the duplicate row. If your data is inconsistent, you'll run into problems analyzing it. For example, if you're trying to run an analysis on numerical data, but a column you're working with has some string elements, you'll get an error. You need to determine the appropriate data type in that, in that case and convert all the data to the same type. You also want to make sure your data is consistent in other ways. Such as having the same format for all column labels. Finally, you may run into a situation where you need to handle outliers. These are extreme values that are significantly far away from the mean of the column. You need to determine if these data points should be included in the analysis or removed. There will be other problems you will run into and you'll learn a few ways to address them. Keep it in mind, it will be important to communicate when cleaning data to figure out exactly what's going on in your dataset. But for now, to sum up, data needs to be cleaned before it can be analyzed. Make sure to check for missing data, duplicate data, inconsistent data, and outliers each time you get a new dataset.

****

Video 2: Inspecting Your Data (4:39)

Before cleaning your data, you should explore. Let's use a subset of the social media dataset for this lesson and look at some methods and attributes you can use to understand your data. First, we'll import social.csv into the data frame which we'll call social. We want to see what our data frame contains. We can peek at just the first two rows using the head method. Just type the name of the data frame followed by .head in parentheses. Here, you can see the first five rows of the data. This gives us information about the columns we have in the data and can help us identify potential problems for cleaning. Here, we see that each observation has an ID number and a response to various questions asked in the survey. If you look closely, we already see some problems. For example, ID number four is entered here twice. We need to remove one of these rows to avoid having duplicate data. We also see some missing data. For example, for ID number two, data for intfreq, home4nw and several other questions is missing.

We'll need to figure out how data was collected to see what we can replace the missing data with. In addition to using the head method, we can also use the tail method. This will give us the last five rows of the data. Both the head and the tail method have a default argument of five for the number of rows it shows; however, if you want to see any other number of rows, we can simply answer the argument in parentheses like this. Let's now look at a few attributes of the data. Recall that attributes give you information about characteristics of an object. Unlike methods, they're not followed by parentheses. An important attribute to know is columns. Using head or tail, we can see all the column names by using .columns lists the column names out neatly, so you don't have to scroll across the data frame. Here, we see all the columns of the social data frame. The column names here refer to the questions in the survey. To see which questions each column refers to, we need to go back to the survey and map the column name with the survey question. For example, if you want to know what intfreq means, we can open up the document with a survey. We see that the question entry asks about how often do you use the Internet.

We can also see the responses here are one for almost constantly, two for several times a day, three for about once a day, four for several times a week, five for less often, eight for don't know, and nine for refused. This information will come in handy for data cleaning, especially when it comes to missing data. For example, if we have a blank for a certain participant, we might want to replace that with a nine since having no response is identical to refusing to answer the question. Additionally, this is important to keep in mind for analysis. When we're finding the mean of this column, for example, we would want to ignore the eight to nines because these would skew the average weigh up and aren't actually meaningful responses to count towards the average. Let's move on to the shape attribute. This tells us how many rows and columns are in the data. Here, we see that there are 2003 rows and 54 columns, meaning we have 54 datapoints for each of 2003 people. Remember, this could include blank cells which represent missing data. Finally, let's look at the info method. This will give us more information on the columns. It tells us the number of observations that are not null for each column and the data type in that column. Here, it has 2003 for all the columns. This is problematic because from our inspection we know that some of the data is in fact missing. We'll need to write some code to identify these blanks as null data points. That's why it's always important to inspect your data before you clean it. While there are several methods and attributes we can use, some of the most helpful ones are head, tail, columns, shape, and info.

****

Video 3: Strategies for Data Cleaning (3:20)

In this video, we'll look at some strategies for cleaning data. We'll create the data frame social one more time and import numpy as np. Let's say we want to find the mean of the intfreq column. Recall that we do this with numpy's mean function. However, here we get an error. That's because the intfreq column imported as a string rather than an integer. Recall that we can use the info method to find out the type of each column. Here we see that the intfreq column is not numeric. It says here that it's an object data type which is a general data type. This usually means it's a string and this probably happened because some of the blanks we have for some rows. To convert it to numeric, we use the pandas function to numeric. We'll reassign the intfreq column of the social data frame using the to_numeric function.

This function takes two arguments: the name of the column and what we want to do with errors. Here we set the errors to coerce to coerce the values into numeric. Now we'll see that the data type here is a float so it's a numeric data type. You'll also see that in place of the blanks, we have some NaNs. NaN stands for not a number. Before we analyze this data, we also want to replace the 8s and the 9s with NAs. Recall that 8 means that the participant said they don't know and 9 means they refuse to answer. We can do this using the replace function. Remember that the replace function has two parameters: what you want to replace and what you want to replace it with. We'll use numpy's NaN for this. To check if this worked, we'll use value counts. Value counts tells you the frequency of each value. So, if we use value counts on the intfreq column of the social data frame, you'll see we now have only the values 1, 2, 3, 4, and 5. The 8s and 9s have been removed. Now we can find the accurate mean of the column. Keep in mind that typically in your workflow you wouldn't run an analysis here. I just did this to demonstrate. In reality, you should clean all your data completely before analyzing it because changes to other columns and rows might affect your analysis here as well.

****

Video 4: Dealing with Missing or Duplicate Data (3:21)

In this video, we'll discuss what to do with missing or duplicate data. In this notebook, we have the code that imported the social dataset, converted the entry column to numeric, and replaced the eight and nines that in column with NAs. We'll now discuss ways to replace NAs with values or drop them altogether. The option you choose will depend on your data and your analysis needs. For example, if we know that the NAs resulted because of a data entry error, where zeros were mistakenly left blank, we'd want to replace the NAs with zeros. Let's take a quick look at our column. To replace the NAs and NANs with zeros, we'll use fillna. In the parentheses after fillna, we want to write what we're replacing our NAs with, in this case zero. Looking at the column now, we see that the NaNs, for example, the one at index one was replaced with a zero. I'm going to rerun the previous code now to reset this so I can show you how to get rid of NAs altogether. If you want to get rid of all missing data, rather than replacing it, we use drop na on either the data frame or a specific column in the data frame. If we want to get rid of NAs in the intfreq column, we do it like this. If we want to get rid of the NAs in the entire data frame, we do it like this. Now let's switch gears towards dealing with duplicate data. Remember that if you look at the head of a data frame, we can see that the data for ID number four was entered twice. This could also be true for other data in the data frame, in rows that we can't see right now. To drop duplicates, we simply use drop duplicates. Here, we get rid of all the duplicate rows in the social data frame. Don't forget to reassign this to social in order to save changes. Looking at the head of the data frame now, we see that the duplicate for ID number four was removed. And now there's only one row for ID number four. If you want to see how many rows got dropped, we can use the shape attribute. Before we had 2003 rows and now, we have 2002, so only one row got dropped. That's all for now. I hope you enjoyed this brief introduction to replacing missing data, dropping rows of missing data, and removing duplicates from a data frame.

****

Video 5: Data Cleaning: Wrap Up (1:25)

There's a lot more to data cleaning than making sure that your data is of the correct type and that missing and missing data and duplicate rows are handled correctly, but these should be some of the first and most important things you check. As you get more experienced, you'll need to learn more advanced techniques, like merging datasets and writing your own functions to clean data for your specific needs. For now, though, let's learn how to save and export our clean data so you don't have to clean it each time you read it into a data frame. After you've gone through the extensive process of cleaning each row and column of your dataset to meet your specific analysis needs, you can export it to a CSV using the to_CSV function. Type the name of the data frame followed by to CSV, and in parentheses, enter the string with a name you want to give your CSV. Let's call this one social cleaned dot CSV. That's all you need to do. Now if you go into the directory or folder where you have this particular notebook, you'll see a new CSV file saved that has the updated cleaned data.

****

---END---