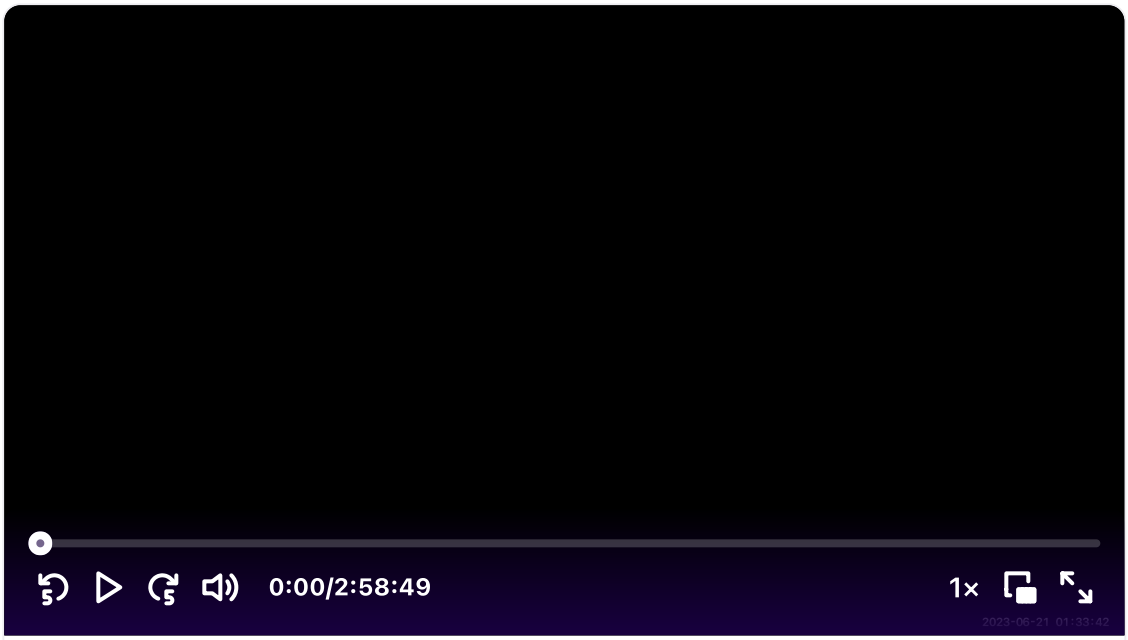
Day 12 Merging and Data Clean Project



Select text to create highlights Learn more

Transcript

S1 Speaker 1 0:01

Yes. So who, who has a question?

S2 Speaker 2 0:05

Oh no, it just says bring up the recording. You're good.

S1 Speaker 1 0:08

Oh, perfect. Yes. So we're gonna talk about that as well. And at the end, at the end, we're gonna see ways on how debug we are gonna learn how to debug. But I'm pretty sure you are debugging right now when doing

your homework, when doing your activities. But we're gonna recommend you a path on how you do it and, and, and don't get frustrated like real quick. Okay. So you get more hands of it. So let me start then sharing my first part of the code, the merging data. Yes. Perfect. You can see my screen. Oh no, it's loading.

S1 **Speaker 1** ▷ 1:02

It would be good. Can you see my screen? So, yeah, perfect. Okay. It's gonna be this one. That one. Perfect. Please thumb up if you can see my computer notebook. Okay, perfect. Did you notebook you can see. Okay guys, so we're gonna talk about merging. Maybe this concept is new for you or maybe it's not new for you. But basically what we are gonna, what we're gonna talk is about this really, really cool function that you, you can use with pandas. You can merge without pandas. Yes. But pandas is like really fast on doing this part, okay? And one thing that we have to understand today is that as, as, as an analyst, we get often different data in multiple tables, right? And it's difficult to work with different data in a bunch of different tables and it's not ideal as well. Okay? Well, in pandas we are going to easily combine or merge separate data frames or similar series or values or whatever, or however you want to see it with the merge method, okay?

S1 **Speaker 1** ▷ 2:41

What you see here in the first chunk of data is we basically are creating a data frame here. Okay? Data frame, remember it's a dictionary. In the dictionary, I have a list and I have a key, key, key and release as values. Okay? I am creating my data frame with my columns customer id name and, and email how with PD data frame. And I can, and you can see here this data, I'm gonna create another one, okay? With another customer id. Customer id. But with this one it's gonna contain item and cost. Okay? So here is the big, the big thing, okay?

S1 **Speaker 1** ▷ 3:31

In this code of block that you're seeing, we are merging these two data frames, but we are merging them by the column ID that they share. Okay? This column ID is gonna be really important soon because we are gonna talk about databases and that's where, where we are going to, to

learn about primary keys, secondary keys and those kind of things. For this moment we are, we are going to focus on the merge function, okay? So the merge function is actually what, what you're seeing here, it contains three parameters or we pass these three, these three parameters into, okay, the two data frames that we want to merge here, you can see info and items, data frames, okay? And then we're gonna pass the column to merge the data frames as Nicole wrote. So on this column, merge this two. One important thing about this is that when you put any data frame, left is gonna be my left table.

S1 **Speaker 1** ▷ 4:50

If you put this one to the right is gonna be my right. Okay? And this is gonna matter when we actually start doing this merge by, by left merge, right merge or, or any other intersection. Okay? So here this table is gonna take only the ones that matches in customer ID between the tables only the ones that matches are going, the one that I'm going to fill here, the contrary I have, I can put it on customer id. How? That's another parameter that I'm passing to merge out joint. Okay? I can do a left joint, I can do a right joint and I can do an inner joint. Basically the one that I just did here is the inner joint. Outer joint is telling bring everything, I don't care. Just put those tables together and bring everything. So whenever the the function comes, something that doesn't match is gonna put an empty value.

S1 **Speaker 1** ▷ 6:00

As you can see here, an A, an A, an A, okay, now I'm going to do it to the left, left point basically is gonna take, my left is stable. That literally it's, it's the first table that I'm writing that is left. And then the other one at the right. Why is it important? Because the left is gonna bring everything that is in the left side and is gonna bring everything that brings from everything that, that contains the right side, eh, and does not match with the left side. It's gonna bring it, it's gonna bring it enter, like you can do it here. Okay? I am bringing everything from the left. From the left. I'm bringing Bob and and the email. But because it's not matching with the right, I am actually leaving the right without any values. So you can imagine with the the right joint, I am going to bring the right values and I

am not going to bring the left values. Question so far, guys about merging?

S3 **Speaker 3** 7:21

Yeah, I, I got a little bit confused about it. Is it bring in, so basically what you're saying is the PD merge is gonna be the info df based on the how is this the right or the left?

S1 **Speaker 1** 7:35

This is the left. Okay, this is the data frame that is here up here. This is my info D f, this is my date, my table, my first date. Okay,

S3 **Speaker 3** 7:45

So if we would go back to that sample, it's going to bring all the data frame, the info data frame and the items on what is it the right, so the right is given the command on which data frame are you bringing first or the information of that data frame contains from that side.

S1 **Speaker 1** 8:08

Good. The right is actually telling me bring everything that is contained in the right table in this table. That is the right one is item here.

S3 **Speaker 3** 8:19

So the how equals right refers to the data frame, not the columns of the specific data frame.

S1 **Speaker 1** 8:28

How refers to the joint that you are doing between those tables? What type of joint you want to do between those tables? I do you want to do an outer joint? Do you want to do an inner joint? Do you want to do a right joint or a left joint?

S3 **Speaker 3** 8:44

Right? So the joint will be specifying what again? Like I, I get what you're saying, what I've been saying, what I'm trying to see is like what is the data that is coming from that, right?

S1 **Speaker 1** 8:59

The joint sample. Okay. Can you see the image? Yes. Okay, so these basically are the same joints, okay? And you can see there, this is the inner joint, okay? You're gonna get whatever it matches between the left and the right leg. Got it. Yeah. This is the other joint, full joint, just everything together. Okay. Yes, that one. Yeah. Okay. Which is the left? The left joint is bring the complete left table and just bring the values that matches with the right. Oh, I see. The right one is the other part. Bring everything on the right table and just bring whatever it ma matches on the left.

S3 **Speaker 3** ▷ 10:02

Awesome. Thank you. Super clear. Thank you so much.

S1 **Speaker 1** ▷ 10:09

Any other questions?

S4 **Speaker 4** ▷ 10:11

Question? Is it possible to specify multiple joint conditions?

S1 **Speaker 1** ▷ 10:19

Yes, but to be honest, it's not going to do it. It's just going to do just the first one, right? Because it's not going to, I mean the machine is going to accept it, but it's just going to merge with the first one. Okay.

S5 **Speaker 5** ▷ 10:41

Hey Hugo, how do you know items data frame is the right and then info data frame is the left.

S1 **Speaker 1** ▷ 10:49

You decide, you decide that when you are actually typing your code.

S5 **Speaker 5** ▷ 10:53

Okay, and what if it's like three data frames?

S1 **Speaker 1** ▷ 10:56

If it's three data frames, you have to convert this join data frame with the merge data frame and then add it to, to the third one. If you want to put the three of them like all together, like an outer journal of the three of

them, you can put it in in data, in different data frames and you can create a new data frame if you want to do it out, right? And you can create a new data frame like this one containing the three tables here. Okay, Satir. Perfect. Any other questions guys? Okay, perfect guys. Okay, great. So let's today, today we, yes, we have the merging census data, merging census data. It's actually connectivity where you are going to merge the two census data that we created in the last class. Then do a calculation and sort the values, okay? And they give you the instructions on what to do here. They give you a bonus and they give you the unsolved starting code for this activity. We have 15 minutes guys, so please start working on this activity here. I am going to open the breakout rooms, four breakout rooms actually and the same, the same dynamic.

S1 **Speaker 1** ▷ 12:34

The same dynamic that you know already you, you can use your rooms, rooms are open or you can ask questions here. 15 minutes guys from now.

S1 **Speaker 1** ▷ 26:23

How you doing guys? You finish. Thumbs up if you finish. Okay. Okay. Oh okay. I can see some thumbs up there. Okay, perfect. Perfect guys, you finished? Great, great because this is what like as well as got reminder from last class from last Thursday as well and well sort values and let me share the, the solution right now. Can you see my screen thumb up please? Perfect. Okay, so what you see here is that we read these two CSB with pandas, okay Pandas, they don't read csb. And then we print the first five of the average and the first five of the two. This is to see that we have these two columns that are actually the same, well that maybe can share information between them because the other ones they are completely different. Okay? So we're gonna merge that two data frames together based on the tier in state.

S1 **Speaker 1** ▷ 27:43

So in order to do that, it was the same thing that I show you. But because we are merging two columns, we have to put them into into a list, okay? So with two square brackets we put the name of the column here and

the name of the column state and then we can see that when you don't put how the default mode is gonna be an inner joint, okay? Whatever it match between these. So I can print the head and I can see that now is altogether because I see the total population column, the total employee civilians and I can see the three columns of my average csb. Okay? And then we start asking things that we saw on last class. For example, create a data data frame that filters when they're talking about filters, basically we think in lock or lock, okay? So what we're gonna do is we're gonna call that census data frame that we just merged dot lock.

S1 **Speaker 1** ▷ 28:57

We're gonna get the census data frame that specific here, that column I'm using log because the header is a string and this is equal to 19. Remember I am putting in my log the left side and this is gonna be my column, my yes my row. And then I'm gonna bring everything on my columns, okay? So in my row, just bring everything that 2019 and in my columns bring every, okay, then I print there. Then they ask you to add a new column that calculates the forward rate. Remember to create a new column we call the data frame, where we want to create that new column with this bracket. And then with the name of the column that we want to put, remember this name can be anything you like. Okay? And what values we are going to assign to this new column. Well we're gonna assign the census 2019 data frame that we use create. And because we're gonna do some operations, we are actually going to put here a square bracket, okay? Call whatever operation we want to do with which with any column. Okay? So in this case I'm going to use total population in poverty, total population in poverty. That is this one divided by again I am going to access census 2019 divided by my total population times a hundred because I want the percentage rate, right? So I can see my new column here.

S1 **Speaker 1** ▷ 30:43

Now they ask us to sort to sort that data frame. Basically we're gonna sort over this sorted. The F'S a new variable that I am creating, it's going to be equal to my census 2019 df, this one is a new one, okay? Sort values. And remember in sort values I have to put sorted by, okay, this column average per capita income by county, okay, that is the other one.

And this is a column that I just create ascending fold. Why? Because I want the highest lowest. So I want to be descending, not ascending, okay, because I used two columns here. Remember that I told you poverty rate is gonna be increasing, I mean increasing, as you can see here, average per capita income by con three. You see that this one eh 11, is not decreasing, right? Because it is just taking the first one.

S1 **Speaker 1** 32:01

Okay? The first column then print out the data for the most poverty she can state or territory. So basically I want to print just one row here and all the columns row, okay? So I print zero because that's the most poverty and bring all the columns. So it's gonna be all these ones that are here are my columns and these ones are the values of my column. And if you can see here, I am just bringing one, okay? Year thousand and 19 state Puerto Rico, average media, H by county. And you can go and read and then the bonus part they give you, they ask you for the least poverty, second state or territory within one line of code. Here they asking us for the last row, but because we don't, we are not going to be able to count how many rows you can count them and then just go by the index without any problem.

S1 **Speaker 1** 33:08

Okay? And then just put an I lock because you can do that as well. And then count, count all rows in this data frame and then it's gonna give you the, the total number or you can do it like this. Lock the length of overt sorted that is this one overt, sorted. The length of that minus one. Okay, because I have a, I have a header, so give me all that link minus one. And that's gonna be the, the number that I am going to use here in my row. Okay? The columns again. Bring all my column questions guys, about this activity.

S7 **Speaker 7** 33:52

I have a question.

S1 **Speaker 1** 33:54

Yes please.

S7 **Speaker 7** 33:55

So in bonus, the bonus part, why do you have to put the length minus one and why can't you just say index minus one, index minus one is referring to the last row, isn't it?

S1 **Speaker 1** ▷ 34:08

Yes. Good. Yeah, yeah. Yes, you can do that as well. Here is just another way of doing,

S7 **Speaker 7** ▷ 34:13

Okay. Alright, thank

S1 **Speaker 1** ▷ 34:14

You. But yes, you can do that. The index minus one.

S7 **Speaker 7** ▷ 34:16

Thank you.

S3 **Speaker 3** ▷ 34:18

So I have a question about that too. So when you put the index minus one is it is in the beginning, right? Because that's assuming that it's just from the top to the bottom because that one seems like how is it gonna take minus one not the last row instead of the index?

S1 **Speaker 1** ▷ 34:38

Oh it's gonna, it's gonna give you like the count of all the index, the total amount of index. Oh okay.

S3 **Speaker 3** ▷ 34:44

Yeah,

S1 **Speaker 1** ▷ 34:45

It's gonna go through all the data. Yeah.

S3 **Speaker 3** ▷ 34:47

Okay. So it could be what high say index minus one.

S1 **Speaker 1** ▷ 34:53

Yeah, exactly. No problem. Or you can use I lock as well.

S8 **Speaker 8** ▷ 35:03

I have a question. You go in I number eight.

S1 **Speaker 1** ▷ 35:07

Number eight, yeah.

S8 **Speaker 8** ▷ 35:09

So here after divided, why do you have forward slash there?

S1 **Speaker 1** ▷ 35:17

This one? Oh this is in Jupyter notebook for me to jump to another, to another line, right? Because you know invitation matter in Python.

S8 **Speaker 8** ▷ 35:32

Yep, yep, yep,

S1 **Speaker 1** ▷ 35:33

Yep. So I can leave it in just the same line everything and it to work, but it is just another way of jumping and showing it. Oh

S8 **Speaker 8** ▷ 35:42

That's, that's good to know. Yep. Thank you.

S1 **Speaker 1** ▷ 35:50

Any other questions? Okay, perfect guys. So let's go to the next topic of today's class that basically it's talking about be data, okay? I am actually going to be data, so the be method, okay, you see my screen if you can, perfect. So the be method, place values into groups to enable more big growth dataset customization. And we're gonna talk a little bit more about this, okay? We're gonna use the pandas dot cut when you need to segment and sort data values into beans, okay? This function is also useful for going from a continuous variable to a categorical variable. Basically what you have here is that sometimes you have so many categories that you have to pin them, you have to cluster them together if you want to see it like this. And basically you can cluster or or yes bin continuous variables that are integers into categorical value, giving a value to each of those integers.

S1 **Speaker 1** 37:18

Okay? For example, this 1 0 59 0.9, 69.9, 79.9 89.9 a hundred, okay? When you create another list here and you see that this group names is equal to F D C B A, you are suggesting that this values that I have here are going to belong to one of these group names. Okay? But you are not telling anything to the machine. You are just defining two lists. One thing, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5. Okay? They have to be this one, the beans one have to be, oh what have to be bigger than than the group names, okay? Because you are gonna group them. Remember? So here they define this test score data frame. They define a new column that is called test score summary. It's equal to pandas PD cut. This is the one that is actually doing the magic. And what is Scott is expecting? Scott is expecting basically a test course d f my test course wherever I have this data because remember I'm going to show you I am not going to have only 0 5 9 0.9. No I'm going to have 50, 51, 52 in this column, okay? In test score like you can see here, okay? But I have to pass the teams, this is my group, the ones that I am doing the intervals and my label, you write it like this, labels equal to group name, this is going to be my label labels. They have to be string, okay? Because they are labels and include lowest equal three equal because I want to include zero as well. Okay? If you put faults, it's not going to include zero.

S1 **Speaker 1** 39:43

So let's see what's going on here. When you actually create the beginning, lemme go back a little bit. The, the test score summary is actually going to be helpful for you to index this score summary and then you can actually pivot your table by test score summary, how you're gonna do that with a group by now that everything is B, you can actually do that with a group buy cord, D F group, buy by Texas court to and get me the max. Remember if you are using a group buy, you have to use an operator function, okay? The score d f max and it's gonna get me the maximum. Let me show you now before we jump to that, can you see my screen? My Jupiter? Sorry, yes. You can see, okay. Oh perfect. So one thing that I want you to remember is that not everything is numbers of cars for example or numbers of person.

S1 **Speaker 1** 41:03

Okay? And so in so many data frames you're gonna have so many values that it can become like really difficult to comprehend what's going on. Exactly. Okay? That's why you have bins. So you can start grouping those data if they, if they show certain patterns that in this case we are putting the the grades but it can be anything, right? Grouping of plants, grouping of of insects or whatever by size or whatever you are doing your analysis on. Okay? So here you see that I am creating a new data frame. These data frame contains these three columns and these fibro and I put random numbers here into the test scores. I create my beans, remember creating the beans, I have to put one bigger than my labels, okay? But remember one thing they both are least and you define what kind of list or what kind of data you wanna put here.

S1 **Speaker 1** 42:14

One important thing to remember is that these label groups labels are equal to group names. They, they can be easier, yes because you can convert them before, I mean afterwards, but we recommend it to be a categorical. Okay? Be stream. So that's how actually you define here. If it's a 90, then it's gonna go from 80.9 to hundred and A. Okay? So F is gonna go from zero to 50, 9.9 C is gonna go from 69.9 to 79.9 D is gonna be from 59.9 to 69 point. Is that you guys the beginning part? Yeah, I hear the group I, the group I just getting like the maximum on the test score summary that basically is gonna show all of them, right? Because we only have five data sets and what else is here? I think that's it actually Guys, any questions on these guys? What does group five do again?

S1 **Speaker 1** 43:44

Group five is actually putting me this score by this by the new index for example, and it's getting me the max score on each of the rows. But to be honest here it is not doing anything because I just have, for example for Logan, I just have one one login, right? So that's why I just have 1 59. So the maximum here is 59, but when you get like more data, it's gonna give you the maximum of that specific test score. Okay? Yep. Perfect.

For example here, right? I have eight ways, ways and it's gonna gimme the maximum of this way you see? Yes. So it's only with the,

S3 **Speaker 3** ▷ 44:44

I have a question at the very beginning of your, I think it's no go a little bit. Oh, maybe a little bit down.

S1 **Speaker 1** ▷ 44:56

Yep,

S3 **Speaker 3** ▷ 44:57

That was the test score summary. Oh yeah, the test score summary that's defining a new data frame, right?

S1 **Speaker 1** ▷ 45:05

This is defining a new column. Remember when we put yes ze frame and then the square bracket and this name is random, you can choose any name here and it's creating a new column.

S3 **Speaker 3** ▷ 45:18

So that is the combination between the beans and the bean beans and the group names, right?

S1 **Speaker 1** ▷ 45:24

Exactly. That's using with panda. That basically is beaning.

S3 **Speaker 3** ▷ 45:33

Got it. And so the test score is what? Oh yeah, I see the test score. Got it. Oh, I can makes sense now. Okay,

S1 **Speaker 1** ▷ 45:42

Thank you. Any other questions here guys? Bean, go for it. Okay, perfect guys, for this activity actually we have more time. It's actually doing moving rate beaning for this activity we are going to spend 30 minutes working on this one. It's individual. You have to go to any breakout if you don't, if you need help from anyone. So basically you will test your Bing skills by creating things for movies based on their I M D B user bro account one heads up on this activity guys, do you, there is not even in

my solution, there is not a correct bin. Okay? All bins are suggested as soon as yours work is correct and as soon as you show whatever you want to show is correct. Okay? Because we have different solutions here when doing the beginning, lemme finish really quick, the code and then you can start here. For example, I use the max minimum to get the headers, but here I use this and I use these streams in order to create the labels and the bins. Okay? But you say hu, I create different bins. Okay, but it's working, he's actually doing the work. Okay, that's that. Okay, that. Any questions from the activity guys?

S1 **Speaker 1** ▷ 47:45

Okay perfect. So we have 20 minutes to work in this activity. Please start working and if you have any questions you can go to the breakout rooms. The breakout rooms are open. I didn't close them, sorry for that, but they're open actually. And you can go to record rooms with TAs or you can stay here and ask questions to me as well.

S9 **Speaker 9** ▷ 49:02

Sorry, this is activity four or three.

S1 **Speaker 1** ▷ 49:06

These milagros is activity four.

S9 **Speaker 9** ▷ 49:09

Okay, thank you.

S1 **Speaker 1** ▷ 1:07:25

Perfect guys, everyone is back. So let's review that activity, okay, share my screen. It should be notebook and please tell me if you can see my screen on top. Perfect. Okay, so what we're gonna do here is like the instruction set, instructions set, we, we have to get the IMV course from these movies course csb. So we read this one with pandas. Once we have that with pandas, the recommendation is to actually use this max and the user both count it should be here. Okay? So you can see that the maximum is be 3000, the minimum is 243. So the recommendation here or the B, that this exercise we did here is actually to start with zero and then to jump from zero to thousand and four 90, then from 4 49, I mean

9 99, then I start jumping until I reach two 30, 350,000. Okay? So this is because I have the maximum and the minimum, again, you can do different numbers, you can have different numbers than mine. And if your beaning is working, that's okay. Okay, I did like this because we have the labels of zero to two 4K based on this Beaning two 5k to 4.9 k and so on. Okay, any questions about the bin here guys?

S1 **Speaker 1** ▷ 1:09:42

Perfect. Okay, so now I am gonna print my bins just to see the bin. Okay, I, I don't, this part applies the data and places into bins. It's going to be the same as this one. You can see here, this is this one, okay, is this one. But here I am creating the data frame and here I am just printing the object. Okay? So the, how do I write my cut cd.co movies from data frame? I am going to take my user boat, okay, I'm going to put my bin and I'm going to put my labels, my labels are going to be equal to group labels head and you can see I just printing the first four. Okay? I am printing my bins beans and I am printing my, my id. I am Im bot. Okay? And I can see the categories here, like I said is just to see my bin.

S1 **Speaker 1** ▷ 1:10:51

But here is the one that I'm gonna create a new column. I am going to put exactly the same cut here, my function here and I'm gonna think it I am, I am the user both group. This my new column and, and I do my groups as you can see, 100k to 350 K, 50 k to 99 K, a hundred K to and so on. Okay? So it's working, it's great. I can do something with this, right? I can create my group buy by I my by my new column that I just created and I am going to count this bin. Okay? So I can see that from zero to two 4k I have eight. So now I can see the, the value of this meaning right, right? He's actually is giving me something just like that.

S1 **Speaker 1** ▷ 1:12:00

I can get the average as well of each of the first rating columns within the group by object. So I can put rotten tomatoes. Rotten tomatoes, user meta critic. Meta critic, and Im D. And from that get the means OK from this group I as well. And you can see it here like the whole table. Average, average, average, average with my groups. Okay? I useful information, questions about these guys, this activity, these solutions guys are going

to be posted, okay? So you can follow up if you miss something or, or on the scripting that you want to recheck, you can recheck it. The concept was clear guys, any questions?

S1 **Speaker 1** 1:13:03

Okay, perfect, no questions. We can jump the next topic of the, let's go here and we're gonna do mapping, guide. Mapping basically pretty simple is how you give format to the string, okay? Like you see here, format and log the same functionality by using map. Map basically method, we can actually style the entire column at once. There are a lot of things on how to write them. It's weird because those are the, the, the characters that pandas decide to use in order to create different formats. But the mapping, actually this is a format I have the f the column that I want to format, map format, the string, whatever string I want to format. For example my, my income, my percentage, my whatever format, the shrink I want to put here, letter point, whatever. Okay? And then you're gonna specify that format, okay? So like I said, these are weird is not really straightforward. For example, to convert front, you are gonna use the, this whole thing and you're going enc cross it here where it is formatted three and then dot forward and this big thing where it's gonna just put a dollar sign before your number in income, okay? These, for example, will split the number up so that it uses a common notation.

S1 **Speaker 1** 1:14:58

And whenever you, you find errors, for example, let me get back here for example. You have these, no, let me show you in the, in the right let me show you with the right, I'll show you here. Can you see my screen guys? Perfect. Okay. Here, as you can see here, I am actually going to income and in income I am adding this dollar sign. Okay? I am doing the same thing with cost. And that's it for the moment, okay? With the dollar sign only income and cost. Okay? So I go here and look for income and cost and you see that my NA values as well contains the solar sign here and I don't really want to see that on my table if I'm going to present this table. Okay? So that's why it's important to as well remember to actually use the l a for drop.

S1 **Speaker 1** 1:16:27

Now to avoid the formatting new pad, one other thing, once you do the mapping, once you use map, I mean PD map also you can change the data type of that column. Okay? If that column is an integer and whenever you do the map you can change it to an integer, okay? And you can change it to an object, sorry, to a string. Okay? You can turn it back only if you remove that string because if not it's gonna show you an okay. So maybe you're wondering here guys here, I don't know where I'm going to get this, right? Well if you go to and map function, guess what you have here? The documentation of mapping, okay. And series number there is another one. I a better one than this one. Now let me check because there was another map page formatting, let me check because I, I found a really good page where actually you have like different example, something like this where it give you like in a table and different string examples where you can actually get a percentage. Thomas dots, any kind of strings.

S1 **Speaker 10** 1:19:13

I have a,

S1 **Speaker 1** 1:19:14

Sorry.

S1 **Speaker 10** 1:19:15

So if it formats, it does the number, does it switch into a string? So say for example if it's a, if there's a doll sign in front of the number, is that number now no longer a, a double or?

S1 **Speaker 1** 1:19:30

Exactly. It's no longer a double. It's an object. Okay. Yeah.

S1 **Speaker 10** 1:19:35

Alright, thank you.

S1 **Speaker 1** 1:19:37

No problem. This is actually going to help us, give us the percentage just like that. And this is the one with the comma. Remember I think with the comma dot semi column or everything like that, you can just change this, but I have to confirm that web page and I can share that to you. Okay?

Where you can find all the, like the cheat sheet where you can find all the strings, but basically that's it guys. Any, any questions about the map function? So as you can see here is my column, then map, then the string that I want to put and then do four.

S1 **Speaker 1** 1:20:26

Okay. Questions? No questions. Okay, perfect. So the second part that we're gonna do, I mean the, the other thing that we're gonna do, we're gonna actually do like a real example real life example. Everyone we, you can go ahead and just follow me with the version OK on this part because we're gonna do this clean crowdfunding together, it gonna contain like everything on it and you can follow along this part, you can help me as well. Okay? So the first part is pretty clear to all of us. So we're reading the file and we are printing head, okay, this part, it's pretty clear as well. Any questions about the F column actually is gonna get all the columns for our reference. Okay? Because sometimes notebook you go like this and then you, you can't see all of the columns, okay? So limit as well. So here you can see all the columns and what type of object those columns are. So who can tell me what is this block of code doing? Guys who can help me? Here I am using reduced crowdfunding DF equals DF block and then I'm putting a semi column comma and then a list. What I'm doing there.

S1 **Speaker 1** 1:22:27

Yes,

S1 **Speaker 11** 1:22:28

You're getting rid of some of the columns you didn't want from before like ID and category.

S1 **Speaker 1** 1:22:35

Perfect. Good, good. Cool. Front. Yeah, yeah. Very much front. Exactly. That's what I'm doing. I'm actually do applying a filter here. Okay, filter. Bring all the roles, remember but only bring these columns. Hug. Can I do it with DF and then just bring the columns that I want? Yes. With double brackets here. Okay great. So I can see just the columns that I'm interested in. 10,000 rows. So who can tell me what I am doing in the

next block of I am putting reduce crowdfunding data frame or DF equal to reduce crowdfunding log, I am opening square brackets, then parenthesis and then I am putting reduced crowd funding the earth pledge bigger than zero. What I am doing there. Pledge disease,

S1 **Speaker 10** 1:23:41

You're looking at the reduced data that has at least a dollar pledge to him.

S1 **Speaker 1** 1:23:49

Exactly. Thank you very much for that. Yes. Good. And that's how you, how you write. So, so far I'm just being filtering my tape as you can see here, but I can see now like good information here, right? Because pledge for example, it doesn't contain any materials, contain least one goal. So now guys, what I am doing in number tick,

S4 **Speaker 4** 1:24:26

You're filtering for the country, which is equal to just

S1 **Speaker 1** 1:24:33

USS Jamie. Exactly. Story. So basically I have creating a list of the columns that I'm going to use or see or I want to see here. And then I'm going to create a new data frame with this list of columns that actually does show me those projects were hosted in this. How I'm going to do that again, the clock reduced funding, country equal us and bring the columns list so I can see this. Okay, perfect. Thank you guys. And what I am doing on number seven,

S1 **Speaker 12** 1:25:21

Hey Hugo?

S1 **Speaker 1** 1:25:22

Yes,

S1 **Speaker 12** 1:25:23

Good question. Is there, is this just an example of how we can use a list of columns to retrieve that data but since you've already reduced it?

S1 **Speaker 1** 1:25:35

Yes, exactly. Because I wanna create an specific data frame with specific group of columns where actually I can see an specific filter in this case countries country, us.

S1 **Speaker 12** 1:25:56

Okay, so it's, it is just by chance that you're using columns and not the colon to select all of the the

S7 **Speaker 7** 1:26:03

Columns from the reduced data frame.

S1 **Speaker 1** 1:26:06

Exactly. Instead of me putting all this here, right, I find the list and then I use the name of that list. Okay, thank you. No problem. So number seven, what I am doing here, average donation equal to hosted in this one, taking the pledge column then dividing that pledge column by the hosted in US D. Yes. Back's count. Baker's count.

S1 **Speaker 10** 1:26:49

I got a question.

S1 **Speaker 1** 1:26:50

Yes, yes.

S1 **Speaker 10** 1:26:51

Is this the Python version of the first homework?

S1 **Speaker 1** 1:26:55

The Python version of

S1 **Speaker 10** 1:26:57

The first homework?

S1 **Speaker 1** 1:27:01

The one to be honest. Okay, the the Python version for the first, no, I will have to check that. I think

S3 **Speaker 3** 1:27:08

It is, is the challenge of the Python, right? Isn't it? Yeah,

S1 **Speaker 1** 1:27:12

Yeah,

S1 **Speaker 10** 1:27:13

Yeah. Cause we did it in Excel.

S3 **Speaker 3** 1:27:15

Yeah, it is. Oh,

S1 **Speaker 1** 1:27:18

Okay. Okay, good. So yes, basically what I'm doing here is I'm getting the, the average, okay, dividing click by baker's count because I, and, and I'm creating a new column that finds the average amount pledge to approve. But here I'm just doing this series. One important thing here guys, here I am creating the series. I haven't created the column yet. So in order for me to create the column, I am going to put my hosted in U S D F, I'm gonna to put the name of my new column donation and then I am going to actually put the hosted in d f pledge divided by the baker's count. Or I can actually call this variable here. Okay, so number nine, who can help me in number nine? Number nine I'm using,

S7 **Speaker 7** 1:28:26

You're converting the three values in into float the average donation goal and pledged.

S1 **Speaker 1** 1:28:33

Perfect. Good. And what else? Thank you Yian. What I'm doing here,

S7 **Speaker 7** 1:28:38

You're formatting and you're formatting it to two points I think after the decimal value?

S1 **Speaker 1** 1:28:47

No, with map.

S7 **Speaker 7** 1:28:49

Oh, I'm sorry. Okay. Adding the dollar sign.

S1 **Speaker 1** 1:28:54

Yes, exactly. Okay. That yes, I am adding the signed here, I am giving format to that and I am adding the dollar sign here as well and here as well. Okay, so I can see now my dollar sign here. Okay, calculate the total number of backers for all us check. Basically I just have to do on the baker's count here and then put sum and I have the total number then get the average as well. So I'm gonna get the average as well. And for the final part, I'm going to collect those US campaigns that have been picked as a staff pick. So here I am using bullion right by staff data frame, again referring to my old data frame. It's not old but it's this data frame log using my log filter, post it in the UF page equal two and then I can print it. Ok. And then I'm doing a group I the outcome. Outcome, okay. And I am whatever I have, okay, outcome, cancel fail. And so lemme ask you one question here guys. Someone mentioned that you have this same pint conversion in a challenge with Excel, something like that. Yes. No, yeah.

S8 **Speaker 8** 1:30:54

Yes. The first homework was in Excel. We had to count, we had to do the same exercise where you wrote now a code in Python now. So,

S1 **Speaker 1** 1:31:05

Okay, good. And let me ask you a question. What was more difficult with Excel or with Python? What do you think? Excel. Okay. When Excel there, raise your hand if you think if Excel is more difficult. Yes, so yes. Okay, well here with the majority of the people here in this group, at least we can say that Python more intuitive. And like we said in the beginning when we were working on bba, it's easier, it, it's more in good like we mentioned. And that's why, I mean maybe if right now it's a little bit confusing, but you're gonna get familiar with this and it's gonna be more natural for you guys to do it. Maybe you're, you are gonna be taking out the code all the time, like, oh, how do I put the lock, lock and then parentes and then that's perfectly normal. But then you're gonna grab

the, the, the flow of it. Okay. The questions about the crowdfunding cleaning guys?

S1 **Speaker 1** 1:32:20

No. Okay, so let's get a break guys, let's get a break, 15 minute break and then we can come back. Okay? Perfect. Anyone have any questions? You can stay. If not, let's have a break guys. Okay, guys breaks up and we're gonna start the class, start sharing my screen and please let me know if you can see my screen please from top. Okay, great. So what we're gonna see now is the second part of the class basically is about box fixing here. This part, I'm pretty sure that most of you that work already with BBA that work already with some python exercises and activities. You are familiar with this debugging part, right? Because when something is wrong with your code, you will have to go and check where exactly is the error, what things you have to, to change there to modify and then well, you, you have to research there in order to find the, the error. Well, here for example, we are going to actually give you, let me just, yes, because this is, this looks much better on my jupyter and here I'm going to open this one. And can you see my Jupyter notebook thumb up? You can, yeah. Perfect. Okay, so here in Jupyter, what you're gonna see, for example, is you're gonna see, well, this one is actually correct, this is the table, right? The, the frame. And we have some, some columns there. But if I put the actually one, yes, perfect. If I print this column right and then I see this percentage of this column, I'm gonna see that it's, so when I ask veterans, the percentage don't mean what do you think is the error here?

S1 **Speaker 11** 1:54:45

It's the wrong type because those are objects or like they're strings.

S1 **Speaker 1** 1:54:50

Perfect. Yes, exactly, it's the wrong type. And you're gonna see here that it says value error could not convert string to flow and it gives you the, the percentage, right? Because it's a string and you are trying to get the mean of a float or the mean of an, okay. And that's the main problem. Normally when you see these kind of errors, you go, if you scroll down, you're gonna see like the type error. This is the one that actually you

have to focus, okay? Because it's a Skype error could not convert and then give you the string to numeric, okay? So we have different steps. So the first step in fixing a bug is to keep going, okay? You are gonna see errors all around your code and it's normal, okay? That happen all the time. And they are rarely the end of the world. And basically most of the bot that you're gonna encounter are simple enough, simple enough to solve, okay?

S1 **Speaker 1** 1:56:05

You will, as long as you know where to look, you're gonna find a solution. Okay? So keep going, don't get panic on this, you, you, you will find it. Okay? The second step is actually to locate the error here is cc because you're using Jupyter notebook. So basically you can see the code, I mean the block of code, and then you can see exactly where is the error, right? If you were using like visual studio code, you will have to check even in the error they give you the line where you can find the, the error. Even in vba you can remember about that. And, and then you can see the, the key error, you can see the value error, you can see the typer. Okay? So you have to lock it, that, that error, okay? And the third step basically is that if you find the error but you still don't know what to do, you can just copy this, okay?

S1 **Speaker 1** 1:57:13

Paste it in Google and then try to look for solutions and the web, for example, stack overflow, well use webpage where someone actually is giving this information data frame is doing something and then is giving you kind of the same error that you're getting, right? So the people start answering here, first answer as the console output reveal, there is a problem with the data frame column. Then it give you, so my guess, blah, blah, blah. And they give you some solutions here, right? You know what, try to convert everything to numeric like this down close flows, okay? You can copy and paste that one and then that type an object here. Basically you can change this float, okay? You're gonna replace the percentage by space, the string replacement, and then you change everything to float. And then you get the, okay, so that was like the solution for this problem. Of course that can change, that can be different error. You can get a different message there. And, and

remember, you, you, you want to go through the three steps, okay? Who can remind us the three steps of debugging? Maybe Eva or Eva? Sorry, what was that? Can you remind us that? Three steps of debugging. Do you remember? No, I don't. Sorry. No, don't worry. That's okay. Anyone can help. Eva. This step, first step,

S8 **Speaker 8** ▷ 1:59:48

First step is to keep calm.

S1 **Speaker 1** ▷ 1:59:49

Keep calm, yes. Second step lock. You wanna copy figuring out what the bug is? Yes, exactly. You want to lock it where the bug is perfect. And the first step,

S8 **Speaker 8** ▷ 2:00:08

Google it.

S1 **Speaker 1** ▷ 2:00:09

Google it, exactly. Great. First come then check where you have your bug and then the third party, if you don't know right away how to fix it, Google it. Okay? That's the main thing. I know that some of you passed through this already, so you are familiar with this part. And one thing that at the beginning, this is going to look weird or strange because you're gonna see these kind of messages and it can get frustrated for you. Frustrating and not getting the right code for debugging. But like, again, like I'm always repeating this, you're gonna get used to it, okay? And, and the most you use it the most, you get the feeling of finding the error faster, knowing what's going on with your code, you're gonna get familiar and, and it's pretty, pretty much that. Any questions about debugging? Anything that is in your mind?

S1 **Speaker 1** ▷ 2:01:26

No? Okay, perfect. In that case, guys, as you can see here in our agenda of today's cloud, we actually cover everything. Why? Because here we have time for you to actually, in this activity, start, start giving bugging. Why, why? This is really important guys. I'm gonna talk about my project that I am working right now I'm working for a, a healthcare company that

is based in United States. Well, it's a healthcare vendor. They, they are basically providing software, eh, eh, services, s sass, to other healthcare institutions like scheduling, like crps, all that kind of things, right? And one thing that I'm working right now is in the scheduling part where actually they have tons of repositories, which loads of different functionalities to a different application, mobile application, web application or tablet because it's, it's different. And basically what I have to do is to get different prs th those pool requests from different repositories, download the code into my machine and then start looking at that code and then just debug the code because sometimes the code is not efficient enough or they don't have lambda functions or they don't have functions at all.

S1 **Speaker 1** ▷ 2:03:15

And I have to actually take the code that I don't know, start reading the code, testing the code, and then after that modifying the code. That's not a hundred percent of my job, but it's part of my job. And that's why it is really important for you to start checking to someone else called and then working on it, right? Because, well, you can phase this into a job situation. Most probably. That's why this activity guys, you're gonna get tons of of box here. Your job is to take the application and fix it out, okay? Dig true. The provided Jupyter Noble can attempt to fix as many boxes as possible. There are a lot and the box get harder to resolve as the code progresses. Okay? It's once you have finished. But fixing for some, some additional analysis on the provided data set. What interesting theories and or conclusions can you draw about the box in New York City?

S1 **Speaker 1** ▷ 2:04:19

As long as you keep telling yourself bugs will pop up and you will get more bug fixing practice. Consider possible questions and what additional data you could search for in order to draw for further conclusions from this data. The good news is that we have the rest of the class to work on this. Of course I am going through this like on the last six, seven minutes of the class, but we have the rest of the class to go through this part, okay? We have this solved version. If we, we have the resources and if we open this Jupiter, lemme check if it's gonna open

here. No, it's not opening, so I'm just gonna open it here. We are going to see, right, if you run this, you're gonna try, you're gonna start finding box, okay? Feel free to go and Google it. Feel free to ask your classmate. Feel free to ask us as well, as long as you work and practice on your debug skills, it will be, it'll be good enough. Okay? Questions on these guys?

S1 **Speaker 1** ▷ 2:05:44

So basically we have, it's, it's right now eight 40. We have like 30 minutes to work on this. If you finish, again, if you finish early, you don't have any more questions, you don't want to see the solution, feel free to to just go, okay, finish here. If not, stay here with us. We're gonna share the solution and, and then we're gonna answer as many questions as you as well at this time as well. Guys, if you have other specific questions that you want to resolve with TAs as well and you feel confident on this, it's time. I mean, you don't have any more questions. You, you finish your, your work and you understood everything, but you still have questions about the homework, right? So feel free to ask them, we can jump with you into a breakout room and then go through that so you can finish early. Okay? Any questions, guys?

S1 **Speaker 1** ▷ 2:06:53

Questions? Questions? Okay. So let's work on these guys. The rest of the class rooms are open if you want to jump into the room or if you want to stay here with me and ask me any questions you have, okay? Okay guys, I'm gonna actually gonna share the solution of these activity. So for you that stay, how was it? Anyone finish the activity comes up. Yeah, perfect, perfect. I see someone. Yeah. Great, great guys, congratulations on that. So I'm gonna share my screen and here's my screen. So I am going to start running my first, can you see my screen on top first off? Yeah. Perfect. Okay, so running my first piece block of code is telling me name PD is not defined, right? So I have to add another, another cell and I have to put here import and that as P right running this.

S1 **Speaker 1** ▷ 2:51:13

And then he's gonna tell me another error going download switch file. And I'm pretty sure that it's because I'm getting this holes and that's it,

right? I have my first part running because of the deer three and that's great. The second one, it's gonna give me another box. Box. The F is not defined. So here, if I check my box equal to box the F bill. So basically I'm not defining my data frame here, right? I have to define my data frame, how and here, right? Because I read it but I didn't compare this into a data frame. So in here, alright, I define my variable here and then if I run this one gonna give me a different error not in index, right? We invested doing a unit. So basically this does not exist right here. So let me print here, box the F and I can see the columns right if incorrect, this one is the one that is actually complaining about this one.

S1 **Speaker 1** ▷ 2:53:21

Okay? So I have to see this part. So you get pretty much what we are doing here. Okay? So basically we import pandas, we define the box F then columns, okay? In order for me to actually write the, the, the right columns here and re-index the right columns here. So I can have all my column names here and I don't misspell any string so I can rename the, the, the specific name because it's not, it's giving me a puff so I can re rename it and then I can put it here. I can see the new, the new name of that column. Then I can, in order to extract, remember I have to create a or or from where I want to strike that information as type date time, I have to change it because if not it's not going to work, it's gonna gimme a type error.

S1 **Speaker 1** ▷ 2:54:46

Then I can get here the year because DT is actually the one that is gonna get me the specifically the year on my daytime. And that's how I work on, on daytimes, right? And then I can see my type here, my feeling date is the daytime and I can go through all the, the types that I have in my columns. After that I is gonna filter to only buildings with in tested unit greater than zero. If I can do the block data, frame my box F block and then greater than zero in the row and bring all the columns, I can keep going, drop the NA in order to clean the number then count. So I can see all the numbers are metric now. And, and I can actually transfer my postcode into an in. So I can work with the postcode or to a string.

S1 **Speaker 1** ▷ 2:55:56

I can then get the percentage of unity invested by dividing these two columns, printing the head. I can get the mean now because my percentage units invested now is can, is blocked. Okay? I can group iger as you can see here and then value count only the borough and then print theEnd and I can rename that borough to total building infestation. And as well I can do a group by by the borough and by the year and bring these specific columns and get only the zoom of those columns. And as well here I can find the total unit infestation and reinfestation by year, doing a grouping by by year. And then summing these two specific columns and, and then getting the total and as well renaming those columns at the end I will have to merge the infestation by year growth and join to the total infestation in totaling units in years. And I'm gonna merge that with my year column on, on this specific column and these two tables here number, okay, I mean these two, these two columns, this table and join to this table. Okay guys, I was seat, any questions on the debugging part? These basically sums, sums it up. Everything.

S1 **Speaker 1** ▷ 2:57:47

Okay, great guys. Well thank you very much for today's class guide. Well really nice to have your attention and see you on Thursday. We're gonna start a new topic on Thursday. We're gonna start with the first lesson in, in, in one visualization guides. We're gonna see some visualization with Python and you be the notebook. So thank you very much and see you on Thursday. Guys, have a great night. If you're gonna stay, stay, don't worry about it. We, we have the breakout rooms. I'm gonna open the breakout rooms. You can go to the breakout rooms with the TA or you can stay here with me to ask some questions, okay? We have, we open office hours. So thank you very much you guys and have a great night. We are going and see you on Thursday.