Day 18 APIs and Geospatial Data

Summary

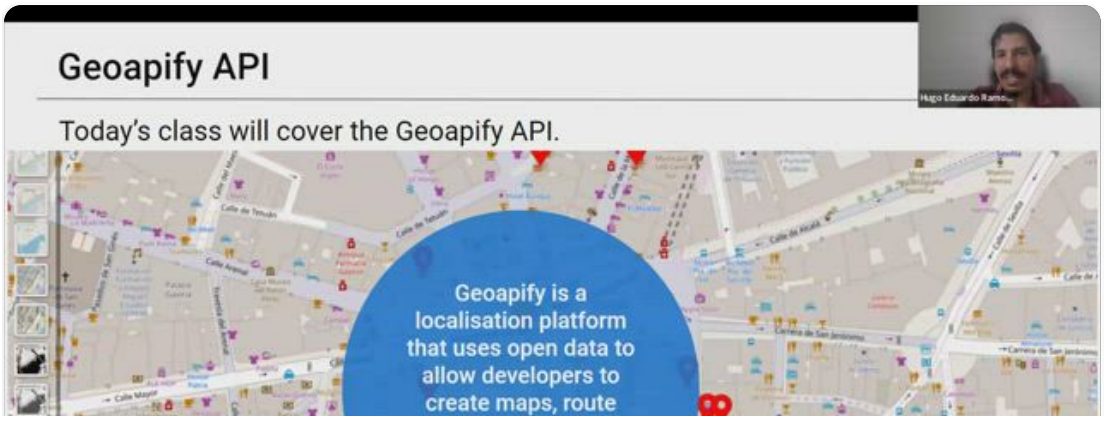
- The speaker is creating a configuration file and saving it to their computer.
- The speaker is demonstrating how to create a m the A P I and Python.
- The ADOCs website provides geocoding and restaurant information, and can be accessed through the ADOCs website.
- The speaker demonstrates how to use the Places API to search for restaurants in Brisbane, Australia. The speaker explains how to use the API, including how to set the search parameters, and how to navigate the documentation. The speaker also discusses the importance of filtering and prioritizing results, and the benefits of using the API.

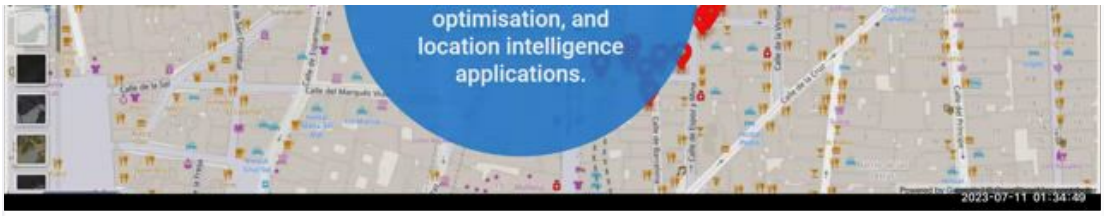
Summarized by AI | July 14, 2023

Summary

- The speaker is explaining how to use the Geo Appify platform, which allows developers to create maps, routes optimization, and location in intelligence applications using open data. The speaker is creating a confi pie, which is a file that stores sensitive information, and is saving their API key in the file. The speaker is demonstrating how to use the Geo Appify platform by creating a Python script that searches for a location using the API key. The ADOCs documentation explains how to use the ADOCs geocode API, and demonstrates how to use the geocode API to find latitude and longitude coordinates for a city. The speaker demonstrates how to use the Places API to search for restaurants in Brisbane, Australia. The speaker explains how to set the search parameters, and how to navigate the Places API documentation. The speaker also discusses the importance of setting filters and using bias to prioritize results. The activity involves creating dictionaries and requests, and sorting the keys by through. The participant is asked to create a code to find the names and addresses of the nearest restaurants with internet access in their city area, but to include conditions.

Summarized by AI | July 12, 2023





Select text to create highlights Learn more

Transcript

S1 **Speaker 1** 0:00

Okay that you are recording that. Basically it is gonna help us for the localization platform that uses open data to allow developers to create maps, routes optimization and location in intelligence applications. So basically these are open apps or open source softwares that are, are actually populated by different data from different places. You have restrictions. For example, the A P I that we're gonna use today can only re you. You can only reach two 3000 requests. That is more than enough for what we are gonna be doing. And of course you can always change your account and get requests if you want. Again, like I explained you the first class, like a premium account, you can get it and then you can get to limited and more features on that specific api. Okay? So another api, but it's more focused on maps.

S1 **Speaker 1** 1:14

Okay. Oh guys, we're gonna do this together. Okay. That's why I am bringing my computer down here in the presentation. The first thing that we're gonna do is I'm gonna share this screen and please guys help me here to actually put thumb up or something like that because we're gonna do this together. I'm actually going

to put this in resources, this link. So please open the link and thumbs up when you finish opening the link. Okay? Unfortunately I can't wait for all of you, but I'm gonna be taking your thumbs up so I can move on. Okay, perfect. Most of you don't stop here. Okay? They have a minority here. Perfect. So you should see something.

S1 **Speaker 1** 2:26

You should see something like this. So if you see something like this in the geo appify, perfect. So the next step that we're gonna do here is that you are going to click in login. Okay? Click login button. Oh, well for me because I'm logged in already, he is, he's sending me with my login, but you should see it like a logging screen. Lemme see if I can show you that login screen. Do you see that login screen? Yes. Okay. So there you will have the options to do it by your Google account or create a new account, clicking create a new account so I can decide which email to use. So whatever you want to do it, but please let me know that you are creating a new account here and thumbs up.

S1 **Speaker 1** 3:31

Okay? I'm gonna give you some couple of minutes for you to create your account. It's pretty easy actually. You don't have to verify your email or anything like that. Password is really open and yes. Questions so far. Anyone is having problems doing these steps? No questions so far. Okay. Jumps up if you are seeing at least my screen ones that you are registered, I mean a, a similar screen on your screen. Okay, I have symptoms but not all of them. Okay? Okay. Okay, perfect. Okay. Who's having problems with this? Please speak up so we can help you.

S1 **Speaker 1** 4:46

Okay, I can see more. So, okay, perfect. Okay, now that you're here, the next step that you're gonna do is you're gonna add a new project. Okay? Add a new project and you are going to name it as followed. I am actually gonna send you the name, I mean you're gonna name it however you like, but I am going to, to give you a suggestion that I'm gonna put in resources. Okay? So it's a data bootcamp them, okay? Okay. Once you do that, you are actually going to, you're gonna get this screen where you, where you can see your API key.

S1 **Speaker 1** 5:56

Okay? Comes top if you see your api I key already. Good, perfect guys. Okay then I am going to stress you about this part because it is really, really important. I am going to save this key, this key in a confit pie. How I am going to create that confi pi, simple as in my case, can go to visual studio code because that's the one that I like to use. Okay? And then here, wherever I have my project, let me start sharing my whole screen and using my screen on top. Yeah, perfect. Okay, so I have my BS code open and here I am going to my desktop. Here I have my folder, then my lesson plan, then I go to Python's API day three here, my activities in my activities. For the first one I am going to create in one of my folders. It's not to be this one, one file that is called config pie. Again, I can create that folder here. I'm in that file here, okay? I am going to open an activities, python, APIs, the three activities, okay?

S1 **Speaker 1** 7:47

And here I am going to folder number one, the 12 one and I can

create in here a new document. This document is a conflict pie. The only thing that I'm gonna put here is my name of the name of the variable and my key. Okay, thumbs up if you, if you finish doing that. Yeah. Okay. So that's it is the other part that I'm gonna be doing is I am going to go back here to my Jupiter and in my Jupiter in my folder I have my config file. If I open my config file, well I didn't separate here, I separate it inactivity number two. But here I can save it this like this, copy this part, this is my config file and I can just save this.

S1 **Speaker 1** 8:54

Okay, thumbs up if that's clear. If that's good. You have any questions on that? Perfect guys. Okay, so in that case I have my first, let's say this like code demonstration here. So the important part here is that we are going to learn how to use that A P i. Okay? We are pretty familiar with this and importing two models. One is request, the other one is Jason, the other one is I importing my confi file, my confi pi. And from there I am actually importing this key. So basically I am going to define a variable in this case target. Target city, it's equal to city Australia. And I'm gonna build an endpoint, a url, target url and I can do it with the F three. Okay? So it's gonna be basically the webpage that I show you plus B one geo code search and one query here that is followed by text.

S1 **Speaker 1** 10:14

That's equal to my target C and format is gonna be equal to Jason and the app P key is gonna be equal to my geo. P i key. So my URL is like this, that's one way. The other way actually is that I go to this link. Let me share this link. That is happy docs gofi com playground geo coding. I'm going to post this one here in resources. And here I just need to put this Sydney, Australia here.

So I put Sydney, Australia here, let me just go here and then just copy piece this one, Sydney, Australia. And guess what? It's gonna give me the request with no gs. Have a script. C U R L U R L, PHP and Java. So I click in Python and I have my request.

S2 **Speaker 2** 11:31

Can you go back how you do it? Cuz it chose different screen. It's not giving me that. When you open gfi G P ifi, how do you do that?

S1 **Speaker 1** 11:46

How to go to this screen?

S2 **Speaker 2** 11:48

No, no. The one that you are getting the Sydney, Australia.

S1 **Speaker 1** 11:52

Oh okay. This one is, yes. Is this this link? Yeah. Ok. That I just share in resources. So basically one thing, let me share this as well is the ADOCs documentation. Let me share this as well in the documentation we have different things here, right? We have a p i, documentation, playgrounds, and code come, if I see here is a playground, okay? So what what we have here, for example, we have maps, we have addresses, we have routes, we have reachability and young places, right? And we have different playgrounds here with my titles, market icon APIs for maps, for addresses, routes, reachability places and more APIs here to connect.

S2 **Speaker 2** 13:02

Well how do you get the as the, your key from the not, not from, I know how do you, you get it from, but like what do you do after you get the key to go to Gify and then add it from there? Cuz you

were showing two ways. One from Jupiter and then one from this website.

S1 **Speaker 1** 13:22

Oh okay, this one? Yes. No coding. So the only thing is that you go to this link

S1 **Speaker 1** 13:28

That I sent you. Okay? That is the geocoding. Why? Because here the whole thing, and I was going to mention it, is that we are going to convert an address into a latitude and longitude, okay? And that process is called geocoding. That's why you get playground and geocoding here, okay? And here you are gonna say you're gonna see geocoding here and you're gonna click text, okay? And you're gonna put the, any city you are interested in, in this case, in this example, we're interested in Sydney, Australia, okay? That's why I put here Sydney, Australia. And that's it. I, I click enter, I just go out from the, from that field and then just go down. And here you have different options. Get the request by the url, you R L C U, rl, you have a script.

S2 **Speaker 2** 14:28

Yeah, I don't see that. And I put the Sydney, Australia and I'm exactly on the same website but I don't see that U R L Q url.

S1 **Speaker 1** 14:37

You don't see the request?

S2 **Speaker 2** 14:39

No, I see this says press search button to generate but it I press and and it's not generating,

S1 **Speaker 1** 14:45

Oh, oh actually you don't have to press. Oh I can check, I can check on you. You can share your screen and I can check on you maybe in the next exercise. So let's see what going. Ok anyone, thank you. Anyone else is having problems like Denise not seeing here Python. Okay. Yes Denise, we can check, we can check on that one. No problem. Thank you. Perfect. So yes, that's how I get this url and then well this part, you know it you data, I'm going to store my JS or convert my result into a adjacent and I'm gonna start it in a variable that it's called. I'm gonna print that json. So you see that is not really pretty so I cannot read it that easily. So what I'm gonna do is I'm gonna use my dumps, my JSON do dumps this function that is gonna pretty print my case.

S1 **Speaker 1** 15:53

So what I want to pretty print, I want to pretty print my variable geo data with an indentation of four and the so keys true. So now I can see it like this dictionary where you have different keys in one of the cases is results and in results you have a list with this information. So with that actually I can get my latitude technology latitude is gonna be what my Q data results, that is a list. And in this list I am going to take the first element of the list and I am get, I am getting from here latitude and longitude. These two, okay, once I got that, this one is the, this last part is just pretty printing off what you are, what you want to see. That's why you use a format. And then the target city in this case is gonna be Sydney, Australia. And I am calling the different values for different case. So I am printing it like in a dictionary format here. Okay? And I can see like a city Sydney, Australia latitude and okay so questions so far about that guys. Okay, so in that matter for example here that

we have the geo amplify place demo places demo in this part, basically as you can see here is basically the same. I am going to put the modules of request the module of Jason as well and I am going to request my A P I. Okay, so I requested my A P I here and now that I

S3 **Speaker 3** 18:01

You go Sorry, we, we cannot see a screen. Yeah.

S1 **Speaker 1** 18:04

Oh my god, can you see my screen now? Yes, yes. Yeah, that was weird. Okay, so basically, well I was explaining about this first part I am important request. I am I importing Jason and I am importing the a p i key, okay? In this API key is the same one that I'm using to hit my geo PFI from this webpage. Okay? And I am actually going to see exactly what can I get. For example, I am going to set here the graphical coordinate for Sydney and Australia, the ones that I got from my last coat. I am going to set parameters for the type of place that I want to get in this case is gonna be category catering, the restaurants, conditions, vegetarians and a radius of 8,000. Okay? So in this case, these ones are going to be defined by my api, okay?

S1 **Speaker 1** 19:28

I mean I am going to get what information specifically for my api. That's why I know what to put here. I know which parameters to use. Okay? So in this case I'm gonna to set the parameters for the type of surgery filter is gonna be equal to my F string circle. So look for a circle with this long, this latitude and the radius, this is my radius, okay? Bias again, if proxy meeting is just going to be longitude, latitude limit 20. So now that I have that done, I can set

up my parameters for the dictionary that I am going to be bringing from that information from scratch. So I am going to put categories, conditions, limit filter bias and a p i key and all of them, they have values already in categories. You have a value in condition, you have a value limit, you have a value and so on. Okay, so I can actually go to my base, my base, U r l in this case again taking the the documentation. We are actually in places, okay? Places, API places, detail, boundaries. I go to play playground, I can see exactly where I am right now and I can play with the information as well.

S1 **Speaker 1** ▷ 21:09

Okay? Hmm. Oh I closed documentation. Okay then I'm gonna run that request using those parameters because I want to bring that, the response is gonna be my request dot get my best base u r l and guess what? I'm going to pass my parent. Okay? My parents is a dictionary so that's why I can create my response u rl with all the filters that I chose me. If I go here, copy the url, paste it here, he's gonna give me these case. Okay, so basically is is that, but I am going to prettify that I am going to put it in Jason response. I am going to prettify that again and I can see now much better this json with my json.com. Okay? And with that I can print the name and address of the first restaurant that appears in, in this case because I want the first one, I am going to put zero. Okay? Play zeta features, zero properties name, that's what I am going to get. And the address, that's it.

S1 **Speaker 1** ▷ 22:50

Beto, NSW zip code thousand. Okay, questions guys. So basically here, like the important part on, on this, the demonstration, it's about the api, okay? How you get the, the, the, the

documentation, how you navigate through that documentation. You can do playgrounds here. That is pretty cool. You can read the talks about an specific thing that you want to see and, and, and basically you can get help. Okay, on on on that specific part. So for example, once that I have that I can go through the documentation just just to check like few more things about the documentation. Okay, let me open this part cause I think this is important and it's quite, okay. So I have these docs, right? The the one that I just opened and we have an explanation here. The places API is a powerful tool that enables you to query local points of interest and amenities effortless effortlessly.

S1 **Speaker 1** ▷ 24:27

You can easily search for places in a city within a an specific fire radius of a point and well the whole thing, right? And we have here this authentication and geocoding api, we went through that. But we have for example API reference where you can learn on how you can configure the request U R L parameters. You get the supported categories there you can learn the hierarchy key using the categorization of the different types of places specifically for PA places. You have the support conditions, support conditions, oh yes, support conditions basically is where you can review the amenities that a place can offer to refine a place search specifically. So you do, you do it more fine, fine research there. And finally you have the UR examples where you can see URL search request depending on what are you, what are you interested there. So each of these, if I go to this one for example, this one is actually expecting an input, right? This input is the latitude and the longitude and radius. You have other optimal parameters here that you can see. You have filters as well that you

can use and, and I think that will be, that will be on that part. So basically what we did here specifically in these places is we have the latitude, we have the longitude and we have the ratings, right? And we can we add some conditions here and with these conditions I can get the first name and address of the restaurant that I'm searching for.

S1 **Speaker 1** ▷ 26:32

Okay? So basically any questions right now about the api? You can set filters, you can set like for example one parameter that can be equal to a circle where the circle, it's an area that you define by a the the bias. Remember bias is gonna help us to priorit prioritization of places near to the location that we are looking. Yes, I mean we can talk about more on this but it's as you can see here a lot. And that's a good thing because it's a very wide a p i with a lot of information that is really useful and you can use. So any questions guys? I

S2 **Speaker 2** ▷ 27:46

Have a silly question. How do we know the parameters? I was reading the readme and it doesn't says like if it's going to be a restaurant or what, what are they when it says in the exercise number three, set the search parameters.

S1 **Speaker 1** ▷ 28:03

This one?

S2 **Speaker 2** ▷ 28:04

Yeah.

S1 **Speaker 1** ▷ 28:05

Okay, let's see the, the, this activity, activity number three. In this

activity you will perform an a p I call to the geo amplifi places and geocoding endpoints complete each of the six drill articulated in the code provided feel encouraged to look back at the previous examples but know that you will be, you will have to consult. Okay? Did you check this? The the starter code?

S2 **Speaker 2** 28:41

I, yes.

S1 **Speaker 1** 28:43

Yeah, okay. Well they, they asking you for Brisbane, Australia, so you're gonna be doing quite the same thing that I just showed you and I guess it's here in the case. Oh, the same? Yeah.

S2 **Speaker 2** 29:02

Okay. Cuz on, on just on the repos. It does not show what

S1 **Speaker 1** 29:07

Exactly what, yeah, no that's why they, they bring the, the the result here so you can take that class of reference. Yeah. Okay.

S1 **Speaker 1** 29:19

Okay perfect guys, any other questions before we start the activity? Okay, great. So let's start with activity. The first activity of today's class basically is this one, the zero three. We have actually 15 minutes to work in this activity. I am going to open the breakout rooms again, four breakout rooms that you can go manually and I'm gonna turn on my light cause it's really close, really dark in here and rooms are open. You can go and ask questions there to the ta. You can stay here and ask questions And we have 15 minutes to work in this.

S1 **Speaker 1** ▷ 46:59

Two more minutes guys and then we're gonna start checking. Sorry, three more minutes. Okay people is coming back from breakout room time, time top guide. So it's time to check checking the activity. Who finished the activity? Guys stumped up. You finish. Okay, I see some people thumbs up. Okay, well it was a long, long activity so let me share my screen and can you see my screen comes up? Perfect. Okay, so basically they ask you to create a code to answer each of the following questions, right? And like I show in the activity they, they give you these hint you will need multiple target URLs and multiple a p requests. Ok? That's why we're gonna start like piece by piece. We're gonna define the dependencies first, the ones that we know already request Jason and U A P A Appify. Key first question, what are the geo coordinate latitude and latitude of Brisbane Australia?

S1 **Speaker 1** ▷ 52:38

So this one actually was a little bit of a copy paste of the code that I was, that I showed before. But the only thing that you needed to change is this one, right? The Brisbane, Australia and pretty much these parameters were the same text, eh? Target city API key is my key and, and then my base U R L is gonna be this one, the one that I have been using. And then I am going to do the request. The request remember is gonna be request don't get. And I'm gonna put two parameters there. One is my base, u r base, U r l and the other one is gonna be my pars, my dictionary that it's equal to my, these two keys with these values, okay? And I am converting this into a json.

S1 **Speaker 1** ▷ 53:35

After that I am preprinting these JSON with my JSON response in the four. And I am sorting those keys by through. So I get this printed, I get the features and the features is a list. And in that list I got something that is called properties. And in properties I have a dictionary with latitude and longitude there so I can get them from there. So I do that latitude, longitude response features, get the first element on them on my list, go to properties and get latitude. I can print the results Brisbane, Australia and the latitude damage.

S1 **Speaker 1** 54:25

Okay, second questions. What are the same thing but now asking for the Sydney Opera house. So you just go ahead and look that like that Sydney opera house and this is going to be exactly the same thing and everything here is gonna be the same thing, but you're gonna get the, this one is the same thing but you get the different thing for this result. Now one that we're gonna use a different category, it's find the name and address of a pharmacy in Brisbane, Australia. Basically they give you this, this link with categories how to use, how do you use the categories. In this case it's a pharmacy, so we're gonna use healthcare dot pharma, we have the latitude longitude, remember from Brisbane, Australia, from up here.

S1 **Speaker 1** 55:31

So we can use those. Then we put the category, the bias basically is gonna be my proximity. I am going to limit my results. 20, I can limit it by 10, by five, by whatever I want to do it. And my bias is gonna be proximity to this latitude and this longitude. Okay, setting up my parameters. Remember I set up my parameters here with only two keys and two values. In this case I will put more parameters in a dictionary. 1, 2, 3, 4 in this case category limit bias

and pq. And my request is gonna be exactly the same thing, but my dictionary is a little bit bigger. So when I do a print, print, I get something like this features in the features, I have a list. In my list I have a different element, but in this case I want to get from properties the address line too. And I want to get my name as well. That is this one.

S1 **Speaker 1** 56:46

So I go for that pre print feature, zero properties, name and address, line two, basically I get price line pharmacy in this address. The next part is really similar for from the pharmacy. In this case we're gonna, they give you a radius of five kilometers from the Sydney Opera house. So we're gonna include that radius, right? Everything is the same. This case is a commercial pet because it's pet, sorry, so commercial pet from where I get this information from this link that they said hashtag categories. Mm, I do the exactly the same thing. But in this case I am going to add the filters. My filters is gonna be by longitude, latitude, and radius because that's one that is asking me five kilometers radius. My bias is gonna be exactly the same thing. Longitude and latitude. And the limit is gonna be 20 the same. So here I am going to add categories, limit filter, that, that would be like my new one. Bias API key, I put my base u rl, I do my response here and then I pretty print this part and that's it. It is gonna be the first element. And with that, with the first element, I'm gonna put name and address line two because I'm using the radius as well. So I, I found this pet barn.

S1 **Speaker 1** 58:31

Okay? Now the other part is actually fine. The nearest is to your house here. Basically you will have to use the, your appify yo code

to find your latitude and inify places to find the dentist. So two of them, right? You put your address here, just like that. You put the parameters, text my address and p i key. And with that actually you can get your, and then you do the response, then you pretty print it and then you can get your latitude and long, right? They are here. Once I have that, I can actually put, set up those as my variable. I'm going to put categories in this case for the dentist is healthcare.dentist. And I am going to put my bias and my limit the bias again pro the limit 20 and the parameters is gonna contain categories, limit bias and pt. I put my base url, I I create my response.

S1 **Speaker 1** ▷ 59:44

I print, I print, print my json and I can find with the, the same thing, feature zero properties and name. I find my West Perth dental center or well my nearest dental center. And it's fun for you to do it because it's, it's nice that when you are coding it, you you find out that this actually works and it's update. And for the bonus part, for the bonus part is find the names and address of the five nearest restaurants with internet access in taxes in your home city area. It is quite the same, but we are actually gonna put conditions here that we're gonna read about conditions. And that's it actually because we have the latitude longitude categories is gonna be catering the restaurants and the conditions is the one that, the new one, right? Internet taxes, I do exactly the same thing. Bias limit the parameters.

S1 **Speaker 1** ▷ 1:00:51

I define a dictionary, I create my base, u r l I create my response in adjacent, I pre print that and at the end, look at this because I am looking for the fi, the five, right of the five nearest. So I need to

use a, a loop with a counter. I need to stop in five. So for place in response, future print, restaurant counter plus one because my country is one, so it's gonna be zero plus one one case property, the name place property, the address line two, and my preprint, I mean my, my limit count there, it's gonna be times 20. So I can put this first separate each restaurant, okay, that's just aesthetics. The counter is gonna be counter plus one. So I can keep counting. So this is gonna be one plus 1, 2, 2 plus one, three, and so on. And if counter is equal five, cause remember we just want five. Then break envelope if you can see your restaurant here. Questions guys,

S2 **Speaker 2** 1:02:22

I have a question for the counter. That's because it starts at zero, but it, it does not set the five places yet that you are the, the, the, the activities requesting. So the counter plus one, it's why is it plus one on this first print?

S1 **Speaker 1** 1:02:45

This is plus one because I, I want to put this print right, I want to increase this string and this string is gonna be increased only if I put this plus one because this counter is gonna be changing each iteration. So basically first one is zero plus one, it's gonna be one and it's gonna print one here.

S2 **Speaker 2** 1:03:07

Got it?

S1 **Speaker 1** 1:03:09

Yep.

S2 **Speaker 2** 1:03:10

And then counter encounter,

S1 **Speaker 1** 1:03:11

Exactly Encounter. Encounter is gonna be printing it is gonna increase, increasing my counter directly. This one because counter is my variable defined outside my for loop, it's gonna be counter equal, counter plus one. So in here my counter is zero plus one, it's gonna be one.

S2 **Speaker 2** 1:03:33

Got it. Okay, perfect. Thank you.

S1 **Speaker 1** 1:03:35

No problem. Any other question?

S2 **Speaker 2** 1:03:42

Oh, I do have another one for the limit. When you say 20, is it 20 feet, 20 meters, 20 kilometers 20

S1 **Speaker 1** 1:03:51

This one or

S2 **Speaker 2** 1:03:52

No, the limits up there.

S1 **Speaker 1** 1:03:53

20 results.

S2 **Speaker 2** 1:03:55

Oh, results, okay.

S1 **Speaker 1** 1:03:58

Yeah.

S2 **Speaker 2** 1:03:58

All right, awesome. Thanks.

S1 **Speaker 1** 1:03:59

You can put ten five as many as you want.

S2 **Speaker 2** 1:04:05

Got it. Thanks.

S1 **Speaker 1** 1:04:09

Any other questions?

S1 **Speaker 1** 1:04:16

Perfect guys. Pretty cool. Pretty cool, right? Okay, so let's jump to the next topic of today's class. The next topic of today's class. We have, let me, can you see my presentation? Yeah. Perfect. So here we are going to talk about exploring nearest restaurants in Madrid. Okay? So for this activity guys, for this activity, basically what we are gonna do is remember in last lessons we learned how to make multiple queries and handle missing data using the accept functions in Python. And the list comprehension. Do you remember, do you remember that thumb? Try and accept last class? Yes. Okay, so basically another way of building out an AP data set is to use pandas, create data frames. So for this, we are going to suppose that we are visiting Madrid and we are in one of the most tourist places that it's called plaza, the main plaza. And there basically we, we are going, we want to look for different types of restaurants in no more than one kilometer from, from this, this plaza, right? So now that we have that context, we know actually we are thinking like, oh of course I actually just finished doing an exercise where I look for restaurants near five kilometer,

five kilometers from my place, right? So yes, the answer is yes, we are gonna work in something really similar, but in this case we are going to use panda, okay?

S1 **Speaker 1** 1:06:21

In pandas, as you can see here, we are gonna use this method that is called et rows. The et row basically is gonna return an index number and the content of each row, okay? So you can access those rows by the row, just like that, the num name of the row square brackets, and then you put the column lane and you get the rows from that column, all of the rows, okay? The best part here is that with pandas, you can use the cat method as well that that is gonna retrieve the result. If that result exists, is gonna retrieve for all the iteration that it's gonna be doing. If, if you don't find any result, you're gonna say don't stop reading it until you find another one. And if you keep getting no results, just keep throwing exceptions. That's why we are actually using the try and accept okay?

S1 **Speaker 1** 1:07:28

To to, to not stop the code and bring all the results that you can find in each of the iterations, okay? We're gonna use the lock in order to store our response and to filter and we are gonna use, in this case a CSB that is gonna contain different types of restaurants. We're gonna see that in the explanation. Let me start showing my screen in with code. Is this one, you can see my code now? Yes. Perfect. Okay, in this case I am getting three are the same and I'm gonna read csb. The csb, it's called ethnicities, csb. In this ethnicities csb, in order to read, sorry, in order to read the csb, I just need to put pd, read CSB and the path, okay? And I will get this a small series of data with Heather Ethnicity and I have

Italian Chinese, Mexican, Japanese, okay, so now I am going to set up in that same types data frame, three more columns, one column that is called name, another column that is called address.

S1 **Speaker 1** 1:08:54

And another column that is called this. Here I am leaving empty this, this column as you can see here. Why? Because I am going to populate this information with the EO api. So now I have a, like this data frame as you can see it here, okay? Who here is telling me find the closest restaurants of each type? The plasma pain, they give me actually the latitude and the longitude already, but I can go ahead and look that with my, with the exercise that I did and I showed you before, right where you actually put the name of the plasma major and with the name of the plasma major, you actually can check the latitude and the, okay, so basically here we have geographical coordinates, the ones that I have it already, and I'm gonna set these parameters. Remember the, the activity that you just did, basically I just put limit radius, it is one kilometer.

S1 **Speaker 1** 1:10:10

My filters here guys, my filters, I'm gonna put circle and longitude, latitude and radius, okay? Three of them because it's gonna be a circle and the radius is gonna be a thousand. And the bias, again, the same thing, longitude and latitude. Setting up my par parameters in a dictionary so I can create my compose URL with my base U my base url, everything is set up here, right? With my, with my parameters, with my base u l. So now I am going to start searching in my application. So basically I print starting restaurant search just to title here and here is pandas, the magic of pinedas working through the API or index and road here. Remember I will have my key and my value. That's why I can start my for loop like

this. For index row in the row, I have to use this row function. If I don't use the et row function, it is not going to ate from here. Okay?

S1 **Speaker 1** 1:11:29

What is the format of json? It looks like addiction, like a Python dictionary. So that's why I'm using, okay, so now let's do something inside my loop for each of the ethnicities that I have. Ethnicities are these ones Italian, Chinese, ethnic, ethnic, Mexican, and Japanese. So what I am going to do, I am going to get the ethnicity first, so I can actually put it in my parameters. Ethnicity is gonna be my type F lock index. Index. That is gonna be iterating 0, 1, 2, 3, 4, we have only four. And what I am going to bring from which column specifically from the column that is called ethnicity. Okay? So I am grabbing this part only.

S1 **Speaker 1** 1:12:24

Now I'm going to add the current ethnicity type to the parameters, how paras creating these paras categories, okay? It's equal two F catering restaurant. Remember the category guys, here I am accessing multiple elements in my json, I am not accessing just one element that I just can define category and then catering restaurant and that's it, right? No, I have to gather different ethnicities that are going to be changing by this variable. So my first one is gonna be the catering restaurant dot Italian Catherine restaurant, the second one, Chinese and so on. Okay, so now I am going to do my API request request dot get base url. The pars that I have already, these are my programs that I have already that they are not changing plus my par that I just created par categories. Is that clear guys? Or should I repeat that clear thumb up? If that, if, if that is clear, would you mind, would

S4 **Speaker 4** 1:13:50

Your mind repeating that one more time?

S1 **Speaker 1** 1:13:52

Yes, no problem at all. So I'm going back here a little bit. Here is the part that we are actually are familiar with, right? We have the parameters defined. These one are not going to change, okay? I why not? Because I am not looking by any other places, by any other locations, okay? This is, these are ethnicities, okay? They are changing by ethnic, not by location. The location is the same one. The location is Plaza Spain. Okay? So that's why I am defining this. They are not changing. The filter is the same, the limit is the same, the file is same and the geo geo key is the same. But here, once I define my ethnicity, it is a variable and it's gonna be changing to my index. Creating through my index. So zero is gonna be Italian, the one is gonna be Chinese, the two is gonna be Mexican, this one is gonna be changing by each iteration. Okay? For my first all this is going, all this code is going to run with the first index that is zero. And here I am adding to my par, to my pars, where's my pars? To these pars, these pars variable am adding another key value, my key categories and the value is this one.

S1 **Speaker 1** 1:15:30

Okay? And that's it. That's it, yeah. Perfect. So now I'm gonna make my request restaurants is gonna be the name of my variable equal request dot kit base url and my parameters, I am going to call my parameters and I'm gonna build my URL different times, okay? I'm gonna build my URL in the index zero that is Italian after it finished doing all this. And here I am going to go back and I'm going to put here index one and I am going to build my request

with index one. Then index two, index three. Okay, so I do my requests for multiple index, I do multiple requests here. So I transform everything into adjacent. That's it. I don't need to preprint this because I, I, I know by heart already that I have to call if I want to get the name I want, I have to call my restaurants these requests features in my list, zero properties, name properties as line two and properties. I know that already from from places.

S1 **Speaker 1** 1:17:01

Ok? And here in this try this specific part, what I am doing is actually I am adding that information, those details into my data frame. That's why I'm using pipe pdf f lock. And I am creating a column with the header name, okay? And I am adding this value to that column, okay? Remember if you want to add a column in a data frame, I have to put my types I Z F and I put here name of my new column that is gonna be equal to a value. So I'm doing exactly the same thing but with duct lock, putting the indexes and then putting the name, okay? It can be, look, it can look like a little bit weird, but the same thing doing exact the same and exception. The exception. I'm going to use these two key error and index error and basically again, I am going to check the name and the distance and the address. I don't care because if I don't count a name then I don't have a restaurant and the distance well just do not store any value. Even though if you find a distance, do not store the value on the table. Okay? That's why I'm putting num pa here. That's the only reason I call mpa, num, pa none. Okay?

S1 **Speaker 1** 1:18:53

And that's it. So let me just print the logs, the logs with this F string, nearest type logs, index, ethnicity, restaurants, lock index, name it. I print my, my my, my results and I can see starting with

research nearest Italian restaurant, la su cafe, blah blah. Now let me create from this a helpful data frame. I'm going to sort a result by distance type. DF is gonna be equal to type df sort values by distance. I am not putting a sorting, ascending or descending. Remember default it's gonna be ascending, okay? So whenever I print my data frame, I have this table and then like in 10 minutes, 15 minutes, you call this and you can say to your friends, okay guys, the new restaurant to us is this one LA 70 away from us and it in this history. Alright, questions guys, any questions?

S2 **Speaker 2** 1:20:29

Well, I have a question if, can you scroll down a little bit

S1 **Speaker 1** 1:20:33

Down? Yes. Here,

S2 **Speaker 2** 1:20:36

No, a little bit more down. I oh up, sorry,

S1 **Speaker 1** 1:20:40

Up, up, up. Okay.

S2 **Speaker 2** 1:20:41

Yeah. And there, here at above accept what you do. Try this zero. It's because the restaurant features and then square bracket zero. What is the zero for again?

S1 **Speaker 1** 1:20:54

Zero means that I have a key in my dictionary that the name of that key is features. Okay? That features contains a list, a list of elements. And here I am just interested in the first element. Got in element zero, then I go to the element zero. And in that element I

have a dictionary. That dictionary basically contains these key properties and inside that properties I have another dictionary. And inside the dictionary I have another key that called me.

S2 **Speaker 2** 1:21:38

Got it? Okay, thank you so much.

S1 **Speaker 1** 1:21:41

No problem. Any other questions? Okay, great guys, if no more questions guys, actually we can take a break. Let's take a 15 minute break and then we can come back to see, okay guys break up 15 minutes up. Okay, so we are coming back and we are going to do a, a review of what we just saw. But in this case you're gonna try it by yourself, okay? I am gonna share my screen. I'm gonna go here and can you see my screen? Thumbs up? Perfect. Okay, so this activity is everyone exporting activity, exploring airport, sorry, activity. So in here you can go ahead and, and and join to any of the rooms and try to do it with your partners and with the TAs or you can try it by yourself. Okay? I'm gonna give only 15 minutes for this activities so you can go ahead and try it.

S1 **Speaker 1** 1:38:37

You'll find this activity here, zero six Python three activities and zero five. Okay? So try to work here. When you open the solve version, you're gonna find a lot of information on what to do, okay? Date frame preparation and then give you some starter code here, okay? And then we can review it. So please go ahead and check it out. And then per, if you want, I'm gonna open the breakout rooms. We have the 10 minutes to work in this. Let me open all the rooms. You can join willingly to each rooms or you can just stay here and have questions. If you have questions, check out my

code. You can take the examples directly from there or the script directly from there. You don't remember a specific part,

S5 **Speaker 5** 1:43:56

You go please do you, you mind showing how you got the a p i key?

S1 **Speaker 1** 1:44:05

Yes, no problem.

S5 **Speaker 5** 1:44:07

Thank you.

S1 **Speaker 1** 1:44:09

Okay, let me go here. So basically you go to the web page that I can you see my screen kicking?

S5 **Speaker 5** 1:44:23

Yes. Yes.

S1 **Speaker 1** 1:44:24

So you go to your resources, I left a link. Okay, yeah, that link will take you to this webpage. Okay? In this webpage you will have to sign in or sign up in an icon that is similar, like the one that I'm actually pointing. And then once you sign in, it is gonna give you a page that you have to click register. You're gonna put an email, a password, and once you are register there if I go here, okay, you are going to see this page. In this page you add a new project? Yeah. The name you can put whatever name you like.

S5 **Speaker 5** 1:45:10

Yeah,

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

As soon as you create it, it's gonna show you the api.

S5 **Speaker 5** 1:45:16

Okay? So is the key specific to like the city,

S1 **Speaker 1** 1:45:21

The keys is specific to the API from this webpage, okay. Okay. So these keywords for everything, for all these documentation for

S5 **Speaker 5** 1:45:32

All, okay?

S1 **Speaker 1** 1:45:35

Yeah.

S5 **Speaker 5** 1:45:36

Okay, I get that now. Thank you.

S1 **Speaker 1** 1:45:39

No problem guys, just one more minute we're gonna check the answer. Okay? Perfect guys. So people's coming back, I'm actually gonna share my screen right now so we can see the answer and can you see my screen comes up if you can. Perfect. Okay, so current airports in Australia, we have these dependencies. The ones that we, I talk about pandas, basically I read instead of ethnicity in this case is city. We have Sydney, Melbourne per parents and la and then I'm gonna add the columns that I'm interested in. As you see in the exercise you can see this printout. So basically I'm going to add longitude, airport name and so on all of them. Okay? Now I'm gonna look for geographical

coordinates for each city I am going to find my permits. My permits gonna be my API key and the format which was given and I have my base U rl, I'm gonna start printing starting airport search, same loop for in index row in C speed road.

S1 **Speaker 1** 1:53:59

And then I am going to put city equal row city plus and I'm gonna add this stream that it's Australia to the stream. Okay, this is for the API to find it correct. Then I am going to add the current city to the pars. In this case my my pars E is gonna call X pars text E equal two. That's it. Okay, I'm gonna maybe the A P R request from the base URL plus the pars. The pars, I have three pars in this case API key format and I have the text, I am going to convert all that into adjacent and then I'm going to strike the latitude te ude. This part was given, so the response is gonna be only the results. Zero latitude results zero ude and I am going to log those coordinates. Okay, so I am gonna see searching airport church coordinate for Sydney, Australia, coordinate for Melbourne, Australia. Okay, I have that part done. So I'm gonna retrieve now the airport information.

S1 **Speaker 1** 1:55:14

I am going to put the radius of 50,000. The parameters I'm going to add categories, p i, key and limit. Okay, starting airport details. Search is different from starting airport search because this is the details again, I'm going to do exactly the same thing. Index row, ct, pd, dot et row. In this case C is gonna be equal my row city because my row city now contains Australia. Okay, latitude, longitude, row latitude, longitude, I have it there and I'm gonna add two more powers. The filter and the bias filter. We know the filter longitude, 10 radius and bias is gonna be a proximity

longitude L okay, the base url, I have the base url I am and, and I am going to get a request the base URL plus the para. Then I am going to do my tri except here because if I don't find an I an airport name or, or any variable here, I can keep running my goal.

S1 **Speaker 1** 1:56:26

Okay? So I am going to fulfill this airport name I a t a name airport, the ones that I have here and I am going to be querying it from features, features, features, everything from zero, everything from properties and is going to be changing name, source row, E a t a address line one distance and source again row. And okay, now I'm gonna be printing all these with starting a report. Details nearest April from Sydney, Sydney Kingsburg airport and the airport name. I'm just bring the city and the airport name and then I can print my data frame with all my details as you can see here. Questions about this, activity items, anything that bothers you, everything, right? That's okay. I mean it's this thing about the hitting APIs is practice, practice and practice. You will see it. The beginning is difficult to understand this kind of reeling when we want to get a specific response but then it's, it's like it is, it's a lot of trial and error. So maybe you are not finishing the activities because of that, but don't worry about it. I mean you're gonna, you're gonna get better and better and it's gonna be more fluent for you guys

S2 **Speaker 2** 1:58:18

U those pairs, you get 'em from the docks on the D five, but it was that in the playground or just the dogs in the places

S1 **Speaker 1** 1:58:30

It is. Dogs in the places. Oh

S2 **Speaker 2** 1:58:33

Okay. Thank you.

S1 **Speaker 1** 1:58:34

No problem. Any other question? Perfect. That's great guys. In that case let's jump to the next topic of today's class that is called UVU maps. In this one, UVU maps, I'm going to talk about the maps, let's see if I can show you how the maps work. So here, be honest in this topic, it is really cool because you can see the maps in in Jupiter, but you must have Jupiter version three and the packages that you will install there should be in the right version as soon as you who worked in the installation of geo abuse before this class, okay?

S1 **Speaker 1** 1:59:42

Okay, perfect. So as soon as you work on the installation of this geo abuse, you should be okay, you should be right with the persons and you should be good to run the code and get the maps. Okay? If not for general, I'm gonna show you like the images and I'm gonna show you how they must run with the code. So, so all the class can, can see what is going okay And after that, if you run it and you getting errors, because I think you can most probably get errors because of the versions, we can check it out in office hours individually and we can make sure that you have that functionality working. Okay? So geo use, we are going to go by understanding that it's a Python library that allow us to embed maps directly into notebooks, okay? This library works with notebooks.

S1 **Speaker 1** 2:00:56

If you run your plain python and, and the interpreter for example,

visual studio code is not going to show you any maps. The code is gonna, is gonna run, but you are not gonna be able to see any maps. If you run it in Google collab you will have Google collab is like a notebook, but for Google in in a server in a web server, you will have to do some installations there as well to see it there in your notebook. And what it does is really cool because you can be so like specific locations on data and customize those maps appearance like the circles, like the, the map grids and all that kind of things.

S1 **Speaker 1** 2:01:49

You will have to install it in your virtual environment but not necessarily the recommendation is yes. So you, you can actually work with the environment variable sir. That's why I was saying if June did not install gvu prior to this class, I think in order for us to set up your, your geo views is gonna be, is gonna lay the class and I think we don't have time to go into individual case. Okay? So that's why I'm gonna explain like the whole thing and then in few hours we can go through each of the cases individual. Ok? So in order to install the geo use you will have to install something that is called a HB plot geo use by pipe B and you can install it by conda with conda install slash C I'm gonna put the code there so you can maybe try to resources, okay? So you can check it out so you can install it in your dev environment and that will install the dependencies that you need for for the activity for troubleshooting installation guide. Actually I have here this, this one I'm going to share this link so you can check it out as well and you can troubleshoot that part. Okay.

S2 **Speaker 2** 2:03:42

Will those work also for the Google collaboration?

S1 **Speaker 1** 2:03:49

That one? Yes it should. It should work If you have,

S2 **Speaker 2** 2:03:53

You said that we need to import differently is through the terminal too, right? We don't have to go or

S1 **Speaker 1** 2:04:01

Tori? Sorry, can you repeat that Denise?

S2 **Speaker 2** 2:04:03

For the, for the, because you mentioned that it's that you, we need to import them from for collaboration Google collaboration but that can be done also in a terminal and order and order to be access accessible on the CoLab collaboration notebook in Google, right?

S1 **Speaker 1** 2:04:21

Yes, but you don't have to import it from Google collab. Okay. You can import it from Conda, Conda install this one, Conda install and then you just put it like that in your computer and you can run it in your no,

S2 **Speaker 2** 2:04:34

Got it. Awesome.

S1 **Speaker 1** 2:04:35

Okay. Yeah, perfect. So basically let's go and take out the code here you can see my code, right? I'm hoping you can see the code. Yeah. Okay, so the first part is we need to import pandas and the HB plot dot pandas library. Okay? Guse is G use is built on

whole abuse. That's why we imported the HB plot. We are not fully covering the whole abuse library because it's quite a big library, but it's important to, to know that HB plot had additional plotting capabilities to pandas data frame, similar to map plot leave, but we can create plots with HB plot, remember with PLT and we were creating bar charts and that kind of things. This is going to work quite similar okay. With data frames. So that's why we imported pandas as well.

S1 **Speaker 1** ▷ 2:05:46

And for the scope of this class, we will focus only in creating maps using geo use. But I mean if you say, hey I, I just read the documentation of your abuse and I want to do something really crazy with geo pandas or geo dataset, you can try it. Ok, you are not forbid of doing that. So in this case, like I mentioned, we are gonna use only the whole abuse but you can check the, I mean we are gonna use only the, the HV plots but you can check the whole abuse and geo abuse as well. Okay?

S1 **Speaker 1** ▷ 2:06:27

Sometimes when you are actually working with your abuse, you can get different warnings by Jupyter Notebook and this is because Jupyter Notebook is built in different, with different conditions that warns you about getting a specific data, a specific site of the screen in your notebook of the screens, interiors or whatever. So in order for we to take out that, we are gonna turn up the warning messages there with import warnings, warnings not filter warnings dot ignore. So we actually can, can run our notebook without any problems. Okay? So then the next part is we're gonna define the dictionary that we can get it from Wikipedia but actually is given here the dictionary is gonna call

coordinates and in this coordinated dictionary we will have latitude, longitude, city and population. Okay? We're talking about Australia here. So that's why we are just taking information from these cities in in our, okay.

S1 **Speaker 1** 2:07:41

And you can see the data frame here coordinates this coordinate. The F is gonna be equal to pd.data frame coordinate and I can print my, my coordinate. So we're gonna, we're gonna start creating a simple map like you can see it here, this simple map we're gonna use you of use to do so, and we will use the data frame that we just create and we are going to plot it with a function that you can see there that it's called HB plot points. Okay? This is added by installing the different whole of use and geo views and we are going to create it. You can see that I am going to create a variable there that is called map plot one. It's equal to coordinates D F HB plot points. And I'm going to add there the longitude, the latitude, the geo is gonna be equal true and just a title that is called that is gonna be osm.

S1 **Speaker 1** 2:08:45

Now to display the map, I need to put map plot one. To be honest, I've been having some issues with my notebook lately and, and I need to, when I was preparing this, I need to refresh or, or restart the whole installation in my anacondas and the whole dependencies because Jupiter is not working really good on this machine. I'm using a, a machine that is not mine. So, eh, that's why when I'm displaying this, I'm getting version errors that I need to fix. But here is what you are going to be getting when you put map, map plot one, okay? You're gonna get the map just like that with this latitude and longitude Q2 from this coordinator. We just

defined from that data frame and it is with a big, big zoom, right? With this zoom it's not really possible to see exactly the points that I am putting.

S1 **Speaker 1** 2:09:54

You can see for example here per Sydney, but you can't really see what's going on. That's why we're gonna map with exit site and we're gonna actually do exactly the same thing, the longitude here. But here I am going to frame it with the width and the high OK with high and then with the width and high, I can see this map. Now I can see the whole Australia country, I can see Brisbane, Sydney, Melbourne, and first some of the cities that I actually mark market there that, that, that I market. I can actually start doing some, some customizing to my map, like adding point colors and size. I have my frame width, my frame height and I have two size. Size is gonna be defining by the population. The scale is gonna be 0.01 and the color is gonna be by C. So once I plug that, I'm gonna have something like this by C define and the circle is gonna be bigger or smaller depending on the population that I'm taking on. Ok. So I can, I can change the, the points scholars, the ones that I, I want to see and I want, I cannot, different titles in this case is are in with the same width and height, same city, but it's my, my plot number, number four. Okay. So you can see it here. Now my title is different and my map is like more, I dunno, with the different colors with land, different bright. So that's the different thing.

S1 **Speaker 1** 2:12:02

Okay, I can interact with my chart clicking different points and I can see the information from each of the, any questions guys? On the map? So it's basically how to map and give, basically this is a

part, we create a data frame and in data frame we start creating plot with the map and adding some features to the map and see the changes on the map.

S6 **Speaker 6** 2:12:40

I have a question.

S1 **Speaker 1** 2:12:41

Yes? Mimi,

S6 **Speaker 6** 2:12:43

What is the geo equals true? What? What does that mean?

S1 **Speaker 1** 2:12:49

That we are using the geo geolocalization with this coordinator

S6 **Speaker 6** 2:12:56

And the tiles equal osm. Is this the title?

S1 **Speaker 1** 2:13:02

No tiles see it type of card that you are seeing in this case, this is a tile and when I change the tiles, for example here I choose this tile. My map looks like this.

S6 **Speaker 6** 2:13:19

Okay.

S1 **Speaker 1** 2:13:20

Okay. Yeah. Like this is like in the same like with we can get lot of information about HV plots and all the parameters that we can use in the documentation. Okay, thank you. No, any other questions? Okay, perfect. So we have actually 10 minutes to play around a

little bit with this, with these maps. Again, if you didn't install the geo views, you most probably are gonna have problems in doing this activity. But basically this one, the zero seven S D U airport map and basically you will have to create with you use maps. They're gonna give you some information here.

S1 **Speaker 1** 2:14:25

Okay? Start with code and you will have to use the code that I showed you in order to create the maps. You can, you can do the import and everything, eh, and you can create the parameters even though if you're getting errors for the exercise, you can actually use the parameters. Okay, so let's work in this activity. We have 10 minutes actually to do it and then we can review it together. I'm gonna open the breakout rooms. You can join and ask questions there or you can stay here and ask questions here. Okay guys, we have one more minute and we can check the review, check the answer. So,

S1 **Speaker 1** 2:25:51

Okay guys, perfect, everyone is back. So let me show you the answer and please let me know. Can you see my screen on top? Perfect. Okay, so basically here we needed to do two two plots with HB plot pandas. So we actually import the HBO O HB plot pandas, then pandas pd, the warnings. We are gonna turn up the warnings, we're gonna read the csb, we're gonna read the head of the csb. That is a data frame and with that we can actually start creating a simple map using gvu. How I am going to do it, map plot, map plot one, it's equal to airport cfv plot points, oris, and then I put long longitude latitude equal the tiles. And then I put frames, width and frame height. So my first chart is gonna print like this one.

S1 **Speaker 1** 2:27:05

Okay, I can see that here we see this and, and, and close them map. I just can see the simple map. Then he is gonna ask me, used to create a custom map by setting values for color, size, and, and title different than osm. It's not title, it's tile. So configure the maps, it's gonna map the map plot two longitude, latitude, well the same airport gfb plot point. But here I am going to use a every, the size is gonna be marked by the distance, by the distance here and the scale is gonna be 0.5 and the corner is gonna be city and then it's gonna look something like this.

S1 **Speaker 1** 2:28:07

Okay, question guys, about this activity. I mean once, once it's running this parameters, setting up the parameters is pretty, pretty much reading the documentation and pretty much trial and error on how you want to fill your map because everything is there. Okay, perfect guys. So we're gonna jump into our last topic of today's class. That is actually another way of reading APIs. This is the last way that, this is the last thing that we're gonna, we're gonna learn on reading APIs and it's called us census. Okay, can you see my screen? Sorry. Yep. Perfect. So in this case we're gonna obtain an API key from this webpage ww census gov developers.

S1 **Speaker 1** 2:29:09

This, I'm just gonna share this into the resources, okay? And one important thing that we have to learn here about this specific API is that basically for for any of the US information, they actually any, any national statistic office worldwide actually provides an ultimate way to fetch data such as APIs or endpoints to retreat

data in different formats. And in this case, because is there already we we, I mean Python is embedded with that information. So we can install that information into our system and we can read it directly from the Python library. Okay? So that's why it says here, Ron, keeping salt sensor in your environment. So this is a different way, I'm going to show you the wrapper that this wrapper provides an easy way to retrieve data from 2013 census based on steep code, the state, district or county. But here you're gonna see a different thing and you're gonna see the different pieces of code because instead of having, for example, poverty, poverty count and employment count, you have labels. The labels look like this, like codes, okay?

S1 **Speaker 1** ▷ 2:30:50

But the advantage of having this a APIs and this information is that you actually can create immediately data frames and we are gonna see how, for example, we have a method instead of getting the request and all that kind of thing is this method that started with data from each of the files. Then we can do different evaluations directly in data frame like dividing poverty rate by total population to get poverty rate. And we can actually, with the US census can does not explicit calculate the poverty rate. So that's why we can calculate it directly with our data frame. Let's see the code so you can, oh well let me show you the wrapper first I'm going to show you, send you the wrapper. The wrapper where you can find the wrapper means that this is like the source where you can get all the information from from this census data.

S1 **Speaker 1** ▷ 2:32:00

Ok? I'm gonna send you the wrapper and as you can see here, it is really familiar, right? It's a GitHub. GitHub with a census. We, you

have different, you have one folder, you have different files and you have this rhythm. In this rhythm you can, you can find all the information of that api. So instead of the API documentation, you have this report and it, it's quite simple to use once you get, once you get used to it because it says like install these p install sensor, then you may also want to install complete the complementary library because you maybe are gonna use some information. So just do PIP installed us for usage. For usage you have to put from census, I import sensors from US import state, right? So it's telling you like little by little what to put, what labels to put in order to get the information that you need.

S1 **Speaker 1** 2:33:05

Okay? I wanted to show you this before I show you the code because in the code you're gonna see some weird stuff that you're gonna say, oh, but what's going on there? Why you are writing that weird thing and you are telling me that that's a poverty rate, right? So that's why I wanted to show you from where this is come now let's see the census demo. In the census demo we have import requests in pandas and from census port and we have a map plot here, okay, to create that a upload. Then I'm gonna to from import from confi import API key from this page, okay? The census as well. And I'm going to use both the wrapper and the API from this because they are both connected.

S1 **Speaker 1** 2:34:02

So now this is like the weird part because we're gonna create an instant of that library and maybe this is weird because this is how they write it in the wrapper. So that's how you have to write it because you see we are calling census here, we are calling it from this, from this wrapper. That's why I am storing this into a variable

and I'm writing census par entities, a p i key and the J. So I'm connecting both here. Okay? Then it says retrieve data from us census using the census library. And they give me the reference specifically where to get that information. Like I was mentioning, retrieve the following page to retrieve the Python library documentation. Python library documentation is gonna be in the wrapper like I was mentioning before. If you want to see like a specific things about the labels, you go to this link.

S1 **Speaker 1** 2:35:04

Okay? We are showing you this API because you can find some APIs that are like that. And, and, and if you are, if you want to use it, you will have to know how to use it. For example, remember that I told you that data on all zip codes in, in 2013 AC census is gonna be with different labels, name is name, but different things like zip code, I remember it was yes, population, medium age, household income per capital, everything has a different label. And to get those, I need to write this as easy calling my instance C, this one is the same, the C can change, okay? You can name a different thing here and you can change the C, but this part is the same. Okay?

S1 **Speaker 1** 2:36:03

Now I will put, this is a magic part. I will put everything into data frame just calling my census data because it's prepared to do this, okay? This is like in a dictionary format that the data frame understands and I can put it directly in a data frame. Once it's a data frame, it's pretty familiar for us to actually rename the columns, for example, with the labels, putting the name of the labels to do some, create a poverty rate, doing this operation with my data frame and then configuring the final data frame for rearranging the, then I can actually put the, the data frame display

how many rows I have there and I can display it and save it into a CSV in the format that I want to see.

S1 **Speaker 1** 2:37:04

Okay, guys, questions Now with your time for questions guys. So as you learn, learn here in summary, after these three days of APIs, one of the most important parts, eh, in working with different APIs is the documentation. The documentation that you're gonna be using to get the information that you want, okay? So it's really important that whenever you're gonna work in with the D new A p i, you go ahead and check the documentation, check what information you can pull out from there, how you can pull the information if it's an adjacent format, if you're gonna read A, C, S, B, if you're gonna read an an HTT P or how come you're gonna be reading that information, okay? So we cannot cover like the whole universe of APIs because they have different documentation documents and, and, and it's not possible to go through all of them, but it's, it is, it is a good exercise to have different examples on how the APIs can come so we can deal with them.

S1 **Speaker 1** 2:38:20

Okay, so guys, so if you don't have any more questions here, we are going to actually do our final activity of today's class. We have the rest of the time of for doing it. Actually we're gonna take like 15 minutes and we're gonna leave one minute to show the answer. But basically is this one banking desert. In this activity, your tasks to understand if there is a relationship between poverty, th population, and the number of banks in a given area to help, we provide you with the census data for every U S C code. You will also visualize this data using the abuse as well, okay? And give you the instructions. So again guys, I'm gonna open the rooms,

you can go to rooms or you can ask questions directly here. Recommendation is good to share your screen and go through the activity and the people can help you get it, okay?

S1 **Speaker 1** 2:39:27

And you learn in that way as well. So let me open the rooms, we can start with the final. If you have questions, please let us know guys, just if you want to keep working, yeah, I'm just gonna show you the solution, but I'm gonna post that solution as well. I mean it's gonna be post, so you can follow through that solution because this last exercise actually summarized everything that we were doing within the last week and the past week about linear regression, correlation, statistics, and it's summing on all, I'm gonna explain it like really quick so we can close the class. Okay, can you see my screen? Thumbs up? Perfect. So banking desert, basically we're gonna import, import all these modules. We have census sensor key, the HV plot, pandas, the pd, the request time, linear regression and pt. We are gonna turn off the warnings and we're gonna create an instance like I showed you about the sensors.

S1 **Speaker 1** 2:55:35

Then again, sensors data by these labels getting this with this ACS five do. And then I'm gonna convert everything into a data frame. This data frame is gonna be populated by population, poverty count and zip code. And I am going to actually rearrange the column here, creating a new column that is poverty rate. Okay? Once I have that, I am actually gonna load from the csb, the C bank data, and I am going to do a merch between these two data frames, how pd, merch, and then parenthesis tip code, pd, census pd, how left on tip code. Tip code, okay, now I have that. I am

going to drop now all the missing data that I have in my new census data, complete drop now and I'm gonna transform my longitude and latitude to afloat so I can work with them correctly. Okay? I'm gonna print the head and I'm, I'm gonna use the G, the go views to create my first map, how I am going to use the tile osm, the width, the height, the geo two with the HV plot points, size, scale, color.

S1 **Speaker 1** 2:56:54

I am going to add all that. Now I am going to compute and print summary statistics for poverty rate, how I am going to print the poverty, mean the medium and the mold I'm going to print as well. The mean, the median of the mode of the bank. And I'm going to print the mean, the media and the mold of the population. Once I have all that, I'm going to create a scatter plot with a linear regression between my X value that is gonna be my independent value poverty rate explaining my dependent value. That is bank account, bank account, everything as as flow, creating my slope intercept, R value, P-value, standard error by the linear regression X value and Y values. The regression values is gonna be this formula, the equation of the line, remember, and I am going to actually create the equation of the line to print in my plot.

S1 **Speaker 1** 2:57:49

I'm gonna do a PTs scatter and I'm gonna do my plot of the regression with my annotation line equation, font size, color, my x and Y labels, and I'm going to show this in the analysis. I can see that my air square is really, really small. So it, there is a really small correlation between poverty rates and bank counts, but we have to keep in mind that we plan, consider or the factors such as population or side of the seat. Guys, thank you very much for

today's class. If you still have questions about this exercise or other exercises, stay for office hours. We're gonna open the rooms and we, and I'm gonna say in this room, so we have five room for questions about this topic, other topics in the past as well and, and as well troubleshooting with any problems that you have with your machines. Guys, thank you.

S2 **Speaker 2** 2:58:50

I'm sorry, can, can you just show your screen real quick and where you do the calculation just to see that real quick to

S1 **Speaker 1** 2:58:55

Check? Yes, no problem. Thank

S2 **Speaker 2** 2:58:57

You. Thank you.

S1 **Speaker 1** 2:58:58

No problem guys, if you have to go, thank you very much for today's class and have a wonderful night. See you tomorrow for project For project. We start project tomorrow and for the next two weeks is gonna be project, okay? I'm gonna explain and set up the rules before tomorrow's class, but in tomorrow's class I'm gonna explain the rules as well.

S2 **Speaker 2** 2:59:21

Are we gonna get the, assign the teams tomorrow?

S1 **Speaker 1** 2:59:27

We can get to assign the teams tomorrow, or if you have a team in mind, you can come with your team. That's why I, sorry.

S2 **Speaker 2** 2:59:37

No, you go ahead. Sorry.

S1 **Speaker 1** 2:59:39

Yes, that's why I'm gonna send you before class certain rules so you can start looking on that, but, but if not in class we can check it out, no problem.

S2 **Speaker 2** 2:59:49

Okay, cool. Will you mind describing where you get the calculation from the probably it's really up. It's the public account,

S1 **Speaker 1** 3:00:02

This one, right?

S2 **Speaker 2** 3:00:04

Yes. Awesome. Thanks.

S1 **Speaker 1** 3:00:09

No problem. Perfect guys, I'm gonna open the breakout rooms and office hours to start. Thank you very much guys. Have a great night.