Workshop 2 Accessing remote hydrological data Transcript

15 July 2025, 09:31am

Samantha Rees started transcription

Matthew Fry 0:03
Everybody else here? Yeah, kind of leading team of people working on on these kinds of things on hydrological data sort of systems and data sets and those sorts of things in particular working on this ftri project, which I think you might have heard about a bit about yesterday. Thanks. Over to kit.

Kit Macleod 0:21
Hi everybody. So I work in sort of Matt's team with a range of few kch colleagues and working on sort of you know hydrological data and digital tools to access and use it. Thank you.

Amulya Chevuturi 0:34
OK. Thank you, Matt and Kit, and we'll start the session today. So today's session is on accessing remote data. So this is one of the very major things, very basic, but very major things that we have to do that we have to get data from very different kind of sources obviously.
In this session, we won't be able to cover all the sources, but I have tried to cover it in three different sessions. Oh sorry. Three different sections and we'll go through them one by one. This session is basically you follow along with me. We will take enough breaks.

Brook Legese 1:08
None.

Amulya Chevuturi 1:09
Please feel free to ask questions in between any of the things that I'm running. Happy to stop and answer any questions if you will have any major troubles, we will take you out to a breakout group. One of the facilities just will take out, take you to

the breakout group and we will fix it and then hopefully you can come back and join the session.

Qidong Fang 1:09
Yeah.
Yeah.

Amulya Chevuturi 1:28
This is another session that you can do on your own afterwards too, but I would say feel free to ask as many questions as you can. You can even at the end of the session talk to us as to if you are having actually current examples where you're trying to access data but you're having trouble.
OK, to start off with I am posting the link in the chat for the Google collab research web page. So if all of you could link, open that and I'm going to share my screen now.

Qidong Fang 1:54
Yeah.
Yeah.

Amulya Chevuturi 2:03
So you should have something like a page like this appear or a page like this appear once you go into the link. Once you have that. If you go to open notebook or you go to file and you do open notebook you should.

Qidong Fang 2:07
Thank you.
Yeah.
Yeah.

Amulya Chevuturi 2:22
Have this inset open up there. What you will have to do is again, I'm posting a link in the chat.

Qidong Fang 2:32
Yeah.


Amulya Chevuturi 2:34
So here what you have to do is copy this link that I've just posted of the GitHub repo. After selecting the GitHub tab. So in this inset go to the GitHub tab.

Qidong Fang 2:41
Yeah.
Yeah.


Amulya Chevuturi 2:49
Put in the link of the summer school GitHub repository and hit the search button. If anyone is having problems, please raise your hand or just unmute yourself and speak because we can fix it now and it'll be a quicker fix than you trying to do yourself.

Jay Lindle 2:59
Close it.

Qidong Fang 3:08
Yeah.


Amulya Chevuturi 3:15
OK. Is everyone followed so far?
OK. If not, please shout out. OK, once this opens up, you will see that there are different notebooks that open up. There's a list of notebooks that open up, so notebooks are usually the Python Jupiter notebooks and they are always. The extension is IPYNB.
Is this file different from the previous? I have a. Did you mean the workshop to Brooke?
Yes, I have updated it slightly so it would be better if you just get another copy.

Qidong Fang 3:50
Yeah.

Brook Legese 3:54
OK, OK, OK. OK.

Amulya Chevuturi 3:56
Yeah. Yes, thank you for asking that question. That is a good question. So all the notebooks used Jupiter notebooks always have the extension IPYNB and in Google collapse you can open those.

Jay Lindle 4:00
Yeah.

Amulya Chevuturi 4:14
Once the Python scripts themselves, the normal scripts are usually with the extension dot py, so for today we'll be working with the workshop #2 remote access of data. So if you could click on it, it will open the notebook.
For this one you will have to always, as we mentioned before, you have to copy it into your own drive. Otherwise any changes that you make will not be saved. So you can hit the copy to drive button here or you can go to file and have.
Save a copy in drive options. So if all of you could do that, that would be great and this opens up a new tab with the title called Copy of Workshop Two. I would say close the previous tab.
And that would be you have opened up everything that you need to do for your current workshop. If anyone is having problems, please do let us know. Thank you Tom for posting the chat about the steps.

Qidong Fang 5:02
Yeah.

Amulya Chevuturi 5:17
I'll wait for a second here and before moving on to the next thing, if anyone has problems, please do let us know.

OK. So just just a brief update on how this Jupiter notebooks in Google Collabs work. If you hit this file button here, this shows you the contents directory, so anything that you download.
Can get saved here, so I would say I would keep it open for this current workshop. You can close it by clicking it again, but if you press it, it opens up. I would say for this workshop keep that open because you will see that any files that you'd access or download will get stored here and that's what we are working with.
To run the code, I'm sure because you ran yesterday's notebook. All you have to do is hit the play button here. Another thing is because these are Python notebooks, usually any command that you have to run outside the Python environment would need something called an.
Mark, that needs to be put in, so if you have to install something that you don't install it within the Python environment, it has to be installed outside the Python environment. So you will have to use the exclamation mark in this workshop. Particularly, you'll be using this exclamation mark a lot because some of the downloading happens outside the Python environment.
Where some of it happens within. So we'll run through all of them. So moving on to the basics of the workshop. So we are trying to access remote hydrological data. There are different ways to do it nowadays. So initially it used to be all manual downloading. So basically what you.
You would do is you would download it based on like you go to the website you click it, you download it. Those kind of things. Then there is command line downloading which basically you run it, you run the commands and it will download files. This way was slightly easier because you can do multiple downloads together by.
Links certain commands and nowadays you are being able to access data which is the third way you're being able to access data without downloading it at all. So your Python code will just read this remote repository, get the data that you require, or access the data. But you don't download anything and you just do the analysis.
So it only downloads the bits that you need, but it doesn't even download it in a physical sense. It just puts it in the RAM and then just process the data and you get the output. So actually you're not doing any downloading, so storage wise it's not much. So before we jump into the actual process of this and go through the different.
Sections I would say hit this command. The first cell I would say everyone should run the first cell.

Qidong Fang 8:11
OK.


Amulya Chevuturi 8:18
And once it's complete, it would have installed all the Python packages you need for this particular workshop. We are only installing very few packages, as you see Czar CF time CSCS 3FS.

Qidong Fang 8:18
Yeah.


Amulya Chevuturi 8:33
Because Google collab notebooks already have a lot of packages installed already. But if you're doing this work, sorry if you're using this notebook in an environment which doesn't have the Google collab Basic Python installed, then you can use the requirements dot text file which we have given here. You can install all the net CDF. Sorry all the packages that you need.
And you you can do the installation easily and run these notebooks other places. This percentage percentage capture is basically a way to kind of not have the output shown. If you remove that or comment that you will see the installations being like the package.
Installations outputs that this is installed. This is installed. That kind of a thing. So far currently we don't need that, so we have muted the outputs. I would say I will wait for a minute here for everyone to get through their first cell. I can see there's a question.
I don't look as I can't.

Qidong Fang 9:38
Yeah.


Amulya Chevuturi 9:42
The system input might mess with that.
Um.
Yes, it's localised. OK people have.

Tom has already answered your question, Chris, but one more thing I would add is just in case if you want to do it on a localised server it might be also good to get a virtual environment and you can then install this in a separate environment than your local environment.
So that's a way to do it. We won't go through that in the current workshop, but in case you need information, we can post links. Please ask how to set up and we are happy to do that and kit, thank you for letting us know. Yes, this does take a while. So it might take a minute or two or even 3-4 minutes for you, which is fine.
There's nothing wrong. This is the longest one that we'll take to run. OK, any more questions before we move on?
OK, so for the manual downloading of data. So obviously as I've said before, you can do click and download. That is the most common way to download lots of data which we have done previously. But there are so like specific softwares like FileZilla etcetera which.
And you know you you go to the FTP servers. FTP servers are generally where these data sets are stored, and then you can download multiple files by just dragging and dropping.
Files. So these are specific softwares made to download this kind of data, but nowadays they're gone out of fashion because there are a lot other ways to download data, much more easier ways. But people still might be using those. So if you have, please go ahead and look at the links that I've posted here.

Qidong Fang 11:26
What?

Amulya Chevuturi 11:33
So another way is for example, let's open up a new link. So this is the link here environmental data.gov.uk. This is basically hydrological data Explorer for.
England and they have all of these data sets there. So what usually you do here is you'll click on something. Suppose you want river level data or river flow data. For example, I choose this particular station, I click on it, it gives me which station it is. Then I click on it further.

Qidong Fang 11:49

Yeah.
Yeah.

Amulya Chevuturi 12:07
And then I get the data. You can view view it as a graph which is shown as a graph. Then you can view it as a table. The data is there and the final option is downloaded as a CCSV and when you download it as a CSV it will just get stored into your downloads photo or wherever you want to go to store.

QF

Qidong Fang 12:07
Yeah.

Amulya Chevuturi 12:27
So this is generally how you would download, but usually when you do these kind of things you only are able to download one station at a time and if you download a lot of data it becomes quite difficult. Another way is.

QF

Qidong Fang 12:40
Yeah.

Amulya Chevuturi 12:42
There is also portals like GRDCGRDC is global runoff data centre. They are one of the biggest organisation. Sorry, biggest data repository flow, global runoff data.
And you get data over all this global stations. Obviously there are some serious gaps in the data lengths, but that's beside the points. That's true for most data sets for real flows because.
There are not that many monitoring stations, but you can download by station. So you can click on the stations. This is I think this might be currently download. They did say there was a thing. Yep. So you can download by station.
And.
Sorry, this is taking a little bit extra, but you can download the data that you want or you can also download by sub region. So even though this is a click and download method, this has an option where you can download.
By multiple stations, so you can click multiple station and then hit download and then it takes a while for them to process the data and an e-mail is sent to you and

you can then just download the data from that e-mail. So a zipped folder is hosted on their servers and then you can download the data. So this is the click and download.
Download method usually which is fairly common. I think all of us must have used that for one thing or the other in our life, and moving on to the command line downloading. But before I move on, are there any questions in the first section?
OK. I'll move on to the command line downloading, so here you can obviously command line downloading is where on Linux servers you can write certain commands and you can download data. I mean obviously you can download single datas. We'll show you how. But you can also download multiple.
Files where you have like a list of. Suppose you have a list of web links of data sets. You can download them iteratively but also the great part about this method is you can use parallel computing. Again, we won't go into the detail of that, but you can use parallel computing.
To like parallelize this process. So oppose suppose you have 10,000 files. So rather than downloading 10,000 files iteratively, you can give this command to multiple servers. So suppose you get give this command to 10 different servers so they are downloading.
1000 files at a time rather than 10,000 files. See really. So that's another way to speed up the process and that's where the strength of this command comes in. So here what we are looking at is the demonstration example or is for.
The EIDC centre, so Ukch has a data centre called Environmental Information Data Centre. I think the workshop five are about the fair data principles is based on this. Not the catalogue itself, but like how, how, what kind of data should be stored, how they should be stored, that kind of a thing. But they have a lot of hydrological and ecological data stored in this website, and this is one of the data sets that we are currently trying to download, for example.
So whenever you go to a data downloading website, usually you have an option called cite. This data set always, always. Please follow give proper source and recognition for the data sets that you're downloading. This is very important.

QF **Qidong Fang** 16:34
I.
Yeah.

Amulya Chevuturi 16:37
You should always be sure about the version of data that you're using. Any scientific analysis you should be very sure about citing the references. There will be a paper associated with the data set. Sometimes nowadays data sets come with their own. DL is.
Which is a digital object identifier. I think object, yes. So digital object identifier. So you should use those. It's really great to have proper sourcing, proper versioning of data that you're using, because that's how people will be able to replicate.
Your scientific analysis.

Qidong Fang 17:18
Yeah.

Amulya Chevuturi 17:19
Yes, Brooke, this is data set for UK. Thank you, Matt for answering that. This is I'm assuming the question was for hydrological Data Explorer. This is just an example data set.
That I was showing you. But the Grdc is a global data set so you can get global data from the grdc portal. Examples are given in the notebook so you can click and go to the websites as needed. Another thing sorry yes.

Brook Legese 17:48
Thanks.
I see. I see. Thank you. Go.

Amulya Chevuturi 17:53
OK. Thank you. Yeah. So whenever you go to this website, they each will have a slightly different way of downloading data. So it is good to read the documentation on how to download data. I would say that is the biggest take away. You should take away from this workshop that although we are telling you common.
Ways of downloading. There will be minor differences in practically every data downloading version, so when you go to download your data they will be licencing all that information. It's a good it's a good practise to give like a cursory review. Can you use this data? Is it really available for you to use?

Qidong Fang 18:24
OK.


Amulya Chevuturi 18:32
For your specific usage type. So for example, for most scientific purposes, just research only. Some data sets are free, so depends on the licence of course, but they might not be free if you're using it for commercial purposes. So you have to be very careful about how you're using the data.
And then you have like download data options you can click on it and there is also the command line. It gives you how to actually download and this is what we will go through in this one. This particular data set on this web page actually doesn't.
The need your username password because most data sets will have a login ID that you will have to provide before downloading. Some of them are free. Again, that's a thing that you will have to figure out with that particular website. So moving on to the actual running of the commands.

Qidong Fang 19:30
Yeah.


Amulya Chevuturi 19:31
So W get is the most common command that you will ever use on Linux servers to get data. It is one of the most widely used and I use it very often. Love this command. So I got this link of this data set.
From.
Download this data. If I log in you can go to download the data option and here this is this is what an FTP server looks like. Usually they'll be like this. So there is GB and NI data so they have divided the data into Great Britain and Northern Ireland. I know the.

Qidong Fang 20:00
OK.


Amulya Chevuturi 20:08
The information sometimes is very difficult to understand, so they do have a read me

option of documentation which you should have a look at when you go on this click daily or monthly you can click on monthly and there are the different net CDF files that you want Matt.
In the next workshop, we'll go through with you what net CDF files are, but for now, we're just going to look at them. So for example, these are the file. This is the file that you want. What you will do is you'll right click on it and copy link address right. You'll copy the link address.
You'll go back to your notebook and here you can just copy that file, link any file link. I would say these will be big files, so please don't try to download daily. Go for the monthly file for now. Just because this is just to show you how it is done.
So once you get the link after writing exclamation Mark W get no space between exclamation mark and the command itself. It has to be without the space, then a space after W get and you write the link. So the link usually ends in an NC, so that's a net CDF file. And when you run this.
You will see that it's downloading and it shows you the percentage of downloading, so you can see that the file should appear on your contents directory. It takes a second if it doesn't show up, hit the refresh button and it will show up. So it's a yeah. Sometimes it just doesn't update. It's there, but it doesn't.
Udate O for me. I didn't have to run the refresh button. My net CDF guy got downloaded. I'm done O this is. I have downloaded one single file.
Is everyone run through this? You can either run the command directly or you can download your own. You can copy your own link and paste it here and then see that you have downloaded.
OK, I'm getting some thumbs ups so I will crack on if there are any questions, feel free to stop and ask which link should we copy. OK, you can copy any link from this so OK, so sorry sorry.
Let me just post something, OK, so if you go to the catalogue.
Let's follow again. Go to the catalogue. The link that I've posted in the chart. You'll get to this page, go to download data, go to again, download data, check maybe GB or NI.
Any of the two, because Great Britain on other island, depending on anything that you want click on monthly, please don't click on daily because that that that file would be too big and you'll forever be downloading the data for now and then click on any of right click on any of these links.
And do copy link address go back to your notebook.

Replace this with your link.
And then hit the play button.
OK.
OK, I'm assuming everyone else has done this, so we'll then remove this file because we want your contents folder to be kept empty, whatever the name of your file is.
Please write it here after the.
The the remove options RM is remove command in Linux so for me it's not 1894 it's 1890 now. So then I'll run the command and it will delete the file if it doesn't get deleted just hit the refresh button and then you'll see that it's removed.
What we're what if we want to extract using a shape file that OK, so Brooke, we will be discussing a little bit more about the specific data downloading in the latest section. Thank you for great question, but.

Matt Dalle Piagge 24:20
That that's coming up next workshop.

Amulya Chevuturi 24:20
I would say, yeah, it's coming up. It's coming up OK. So any questions?
OK, so the next command is downloading restricted data. As I said, a lot of the data might be behind login ID passwords like you have to get an account. This particular example that I chose actually is an open data. You don't need login IDs for that.

Qidong Fang 24:34
None.

Amulya Chevuturi 24:51
But many other data sets on Eidc also can only be downloaded if you have login ID passwords another way that Eidc has restricted data is if you want to download the whole catalogue for example.

Qidong Fang 24:52
Yeah.
But.

Amulya Chevuturi 25:08

If I wanted to download this whole data set from 8090 to 2019, I'm going to the Download file. It's it says here when you have to do bulk download options you have to give your username and password so that is.
Again, if you go through the documentation, it will tell you how to do it, so here they have obviously kept like a requirement that you have to have a login ID password. So for that, although we are not going to run this command, I'm not going to run this command because I don't want to share.
Analogue and ID password widely with the participants of this group. Although I trust you guys very much, but I can't share it so this would be the command all you need to do is you have to put in the whole catalogue link as it was given.
And you have to have this flag which says author no challenge and you have to give your username and password. All you have to do is change the username to your username and put your own password and you should be able to download your whole data set. Please do not run this.
This command currently Google collab will not have the space to download this whole data set. Either the processing power or the storage powers. Please don't do it.
So yeah, this is how to download not only restricted data, but on this particular website this is the way to download multiple files.
But that's what this particular website what happens if you want to download data sets which have multiple files but in a very different format. So sometimes what websites do is they will host the files in a sipped format. I know most of us are very used to.
Zip files called dot zip but on on these very large data sets there are other ways to zip files and tar dot Z are another way to zip files. They have like a higher. I think they have a higher compression factor so.
You can download multiple files using the star dot Z dot star dot Z file and you can go into this link and find this data. So what is usually is like you go to this website another demonstration example.
And then I clicked on different files. I think it was hourly pressip. And then when you click on it you get these star zd, the star dot Z files. So we'll see what these files contains because we are going to download them.
So currently if where did it go? Yeah. Sorry if you run this W get command with the tars, DZ file or any of it, you can just right click and copy the link if you want or just run this particular cell because I've already put the web link there.
If you run it, you will see that the tower, these are the tower zip file. I can't say that

word got downloaded, so now I have that in my contents directory. Right now I have it. We have to unzip it. So to unzip it, the command is.
Star minus ZXVFZ is used when it is star dot Z. Sometimes it's just dot R then you don't have to use the Z option in the flag. This is very common Linux commands, but it's used to it's very good to learn these things because you will be using this very. Commonly, if you're downloading data and working with any hydro climate data sets, so when we run this command and you can run it with me once it's run, I would say hit the refresh button and you will see that I have like so many files now.
In my contents directory. So I have downloaded multiple files by just downloading one zip file and then then unzipping it. So as you see I have like loads of file now. So that's one way of downloading multiple files. We will remove all of these files because we don't need them currently.
So I would say run the remove command and then if you hit the refresh button, your contents directory should be empty. Yes. Thank you. Matt has left a comment which is very interest which is very again very important to remember. So please do go see the chat.
I will take a second of break here again. Is anyone having any issues? If not, we will progress to the next step and we will take a break a little bit after this section finishes OK.
So verafat. Yes, please go ahead with the question.

Weeraphat Duangkhwan 30:03
Nothing, nothing, sorry.

Amulya Chevuturi 30:05
OK, that was a thumbs up. I know I make that mistake too. Another way to download multiple files is to kind of have a list of your text. Sorry list of your links in a text file so you can have a text file which has like 123.
Are your different files that you want to download, so your web links are there, so here I this next cell is basically creating that text file. This you can also create on your own by copy pasting but this is just a way to create in Python a text file. So this is nothing exciting. This you can create.
It however you want. If you run this command you see that you get a URL dot text in your contents folder. When you double click on it, you will see that all it has is just the list of these two web links.

Listed in the text file, there's nothing to it. There's more command that I've added here. Basically, just outputs the contents of the text file. Again, this is Linux Common Linux command, just to see what's the contents of a text file.
So nothing exciting, right? So basically we have these two Web links now with W get if you're running a text file, all you have to do is add a flag called minus I and it will basically download.
These two files, after taking the links from this text file. So we run this command and it has downloaded as you see now it has downloaded two different files. It's going through one by one and downloaded both the files.
And if you see these two files should have shown up in your contents directory again hit the refresh button if it hasn't. But yeah, so this is another way to downloading multiple files, but it will be iterative. So if you note it will happen one after the other. So you have to.
Think of ways of how to optimise this. Do you run it on different like maybe you can chunk different file text files do if you have like 100 files you can have five different text files with 20 different files each. Maybe that works for you. So how do you want to parallelize this if you know?
Know how to run parallel commands. You can use those. We are currently not going into that because that's a whole session by itself, but we are happy to help you if you give you some pointers if you want to learn how to do this on your own. And then the final command in this section.
Is basically removing all the data that you have downloaded. So any questions on this section before we move on to the main part of this workshop which is accessing without downloading, so we don't want to download the data we want to access it without downloading or there are bits of it still where you are.
Downloading. But this is where Brooke's question will start helping that you may be able to download subsections of data.
Any questions?
I will give it another couple of seconds before we move on.
OK, so when you're accessing without downloading, obviously one thing is where your Python commands directly read the data, but before that we will go through other ways of accessing data. So one of the other very common.
Data access ways are these data access protocols or servers. They're called like the opened app servers or the thread servers. To be fair though, these have, although they're very common in hydrology, they're not being used as much, but still these

you will find some data sets still.
Using these so I it I thought it would be good to have an intro here on how to download these data sets so I will click on this example Web page where basically NASA data is stored for cmap, 6 for those who don't know cmap.
Six are the model intern comparison for climate projection, so they have multiple models run on the same for the same future projections. So see mix. Six data is the most common data. I'm sure all of you have heard of it or even used it, but if you have follow up questions about that we are happy to answer. But these are very common data sets.
That you would want future climate projection data, precipitation, future projections for your hydrological analysis. So here, as you see the data is stored in different servers so it is stored in this Amazon Web Services.
Which is the simple storage service S3. We will talk about this in the next section. So for now, we're not using this particular server, but another one is the thread servers or the opened app servers. And this is what we are going to be discussing today. So in the open app or not today for this particular section.
For the opened app servers, if you click on it you get like this huge list of links. If you don't know this already, these are all the global models used for the projection. So suppose if you want a file you can use any file. For now I'm using the My Rock 6. This is a global model.
You click on it, then historical and the different socioeconomic pathways, so there are different projections, and this historical data for now I'm just using the historical data. This is the version Ensemble member basically. And then I'm going to PR. So these are like different variables.
I know for people who don't use this data this seems very confusing. I'm just going through it quickly because I use this data very commonly, so I know PR means precipitation. I know task Max means temperature, it's air temperature, maximum daily air temperature. So you would.
Need to know to like go through this but there will be documentation so you don't have to worry about it. So if there is something that you're worried about, just just read the documentation you will find what all of these variables mean. For now we know so I'm taking you through this very quickly so precipitation and suppose I want to.
To download this data set right this particular file, so I will click on it and this takes me to this data catalogue. So there are different servers that you can download this

data via. So there is the open app server, the HTTP server, the net CDF subset server and the WMS.
We go through the first three, we are not going to go for the the the final one which opens an XML file because that's a very specific geoscience data sets. And I we don't particularly use that currently, so.
An HTTP server. When you click on it, it basically if you see it's downloading for me in my downloads folder. So it's directly downloading my file. So for now I'm just going to close it. I'm not going to download it but it's a direct download click download thing.
Right, so you can use this HTP server link, you can right click on it, copy link address and you can use W get on it directly. So that's one way to do it which we have gone through in the previous section. You just W get the link and you download the data. Simple already gone through it. Another one is the net CDF subset. So here what we are talking about now we are coming on to how to subset data. So when we click on this I should have opened a new tab. OK, sorry. I'll open a new tab.
So when I listen. OK. Sorry. Yes, when you open the new tab you have something like this, right? The data set link which you need to download is down here.
And you can subset the data by going into this link and you can say OK, I don't want from 90 to -60, I only want it from equator to 60° N.
Oh, sorry, and I want maybe.
Yeah.
10 to yeah, 10 to. Yeah. For now it's just a very small data set that we need and you can change the time range. You can put it to single time and you can change other options. You can also have different types format, but for now.
We will just leave it at that when you start changing these boxes, you will see that your this is changing. Basically your net CDF file is getting additional flags attached to it. I would say so because I put it from 10° to. Sorry.
W 10° to east and then I have not only 60° and South is only zero time is also single so.
This is a single like a subsetted file, so you're not downloading the whole global data, you're only downloading very, very, very targeted file that you have, and then you can copy this link.
Copy this link and you can take it to your notebook, so here um.
Going back to your notebook, you will see that the first cell I've completely commented out, because this is the direct download option that I just mentioned to

you from the HTTP server. We're not using that. You can do it on your own time, but there's no need. You've already followed along in the.
Previous section. So basically here you will download the data whatever you want from the directly but the whole global data gets downloaded. But if you want like a subset of data you subsetted the data using the net CDF page subset page.
And then you basically what you do is you W get the data and you attach the whole link so you can attach the link here. So I already have a link there. You can change it from the link that you copied from the net CDF subset link.
And there is an additional flag that I have added which is minus O test dot NC. So when you download this file now it has all of this additional I would say flags attached to the name itself.
Yeah.
Keep it simple. What we want is we we are saying to the command now is saying basically download this data but rename the data file into test dot NC. So that is what we have done here. So rather than having that whole weird file name.
We want to get a simple file name called test dot NC. So if you run this command you will see that you will get a test dot NC file in your contents directory.
And now this test dot NC file you can open and you can plot it. So currently the link that I have run is the link that was existing in the notebook. But if you copy and put your own.
Bounding box link. You'll slightly get a different output in the next couple of cells because we are plotting the data, so we are importing the net CDF. The packages that we need to plot this net CDF file.
Again, not going into detail because Matt will be running this session about plotting and visualisation in the next session, so don't worry about these commands. But if you have any questions, feel free to pop it in the session and we will discuss them. If not in this session next session, don't worry about it.
And this we are basically using X-ray package to plot the data. So the bounding box was for India. So this is basically parted precipitation for India for this particular time step. The one thing is the units are very different. This is very.
Common in climate data sets to store data as kgs per metre square per second to convert it into millimetres per day, you multiply it by something called 86400. If I'm right, it's the conversion factor is based on the seconds.
In the day, so you convert it from number of seconds in the day to millimetres per day. That's very common. You'll find this. It is very niche, but it's something like

everyone knows how to do this kind of a thing. But if you don't know, you heard it here now.
But so we have downloaded the success of data and we have plotted it. Any questions so far?
And thank you, Matt and Tom for posting links to help additional information.
Sam, if some of these like the Cmep 6 links, if you could just copy them and you can send them to the participants in the resources information, even the even the equation, I would say the kgs per metre square per second one is that's a very common one for precipitation.

Samantha Rees 43:55
Yes, thank you. Will do.

Amulya Chevuturi 44:04
Which we had to I had to learn by a lot of trial and error. But if you know if you know already, that's great. OK. So once we're done with this, we'll remove the test dot NC file because we don't need it.
But then we would go on to like using open app servers, loading into Python without downloading. So any questions?
OK, if not, I will go back to the.
Catalogue server. Sorry I had to move this because I have this weird teams sharing things. I'll go back to the file. OK, so we're back to this web page again and here now we are going to discuss the file final one which is the opened up servers opened up. Servers can read the Python.
What is the function of X-ray here? X-ray is again we will be discussing this Fatima. Don't worry in detail in the next workshop, but X-ray is a Python package that reads net CDF files. What are net CDF files? They're common.
File formats for climate hydroclimate data sets, and we'll discuss all of that in the next session. Don't worry, you will know what X-rays by the next workshop completion. So when you open an opened app server, you will get something like this. And this is the data URL that you will use.
To read in the data without downloading it, so you're just reading in the data without downloading the data, but opened app servers have an additional functionality where you can subset the data before you access it via Python, so you can.
Have like lat long you can say like 0 or 100 so you can download only certain. I would

say subsets of data sets. When you add this information it gets added to the end.
Of sorry.
Yeah, so it gets added to the end of the link.
Sorry it's here.
And added OK sorry didn't come thing work but not sell the access.
1.
This is not working as expected. I am so sorry. Oh sorry. I was supposed to silence these, so once you select these. Sorry, my bad. Once you select these, then you have the.
The variables get selected, so you can just select them and then it has the different extents of them. So it goes from zero to 1 to 599. So there are about 600 data points in the latitude and then about in the longitude you have.
For about 1400 plus data points so you can subset them, you can just say I only want the 1st 100 longitudes. I only want the 1st 100 latitudes and when you do that this gets added to the end of your link. So you're you are subsetting the data set.
So you're reducing your size again, not global. You're reducing the size and then you can copy this data URL onto your notebook. So I already have a file.
Sorry here in the second link, if you see you have a subset of data set right before the before the subset of data set. I had a cell which I had forgotten where I just opened the whole file. So let's run this command and see what happens.
So X-ray again, as I said, is a Python package which reads in net CDF files. Net CDF files are where the formats which the data is stored, and usually when you read in with X-ray files here I have just read in the data set. I've not assigned a variable, I've not sets F equal to again Matt will go through.
Through this in detail, but it is showing you the different dimensions that the data has already that you have been read in. So you have the latitudes which was 600, you have 365 days of a year because I think it was a 1950 data.
And then you have about 1400.
Longitude data set so you have all of this data, but when you subset this functionality you get.
You get subseted data set, so now you don't have the same 1400. You only have about 100 latitudes 100 longitudes and only eleven time steps. This is again a way to reduce your files.
Size and if you have like if you have another way of subsetting data is you can read in the whole file as in the last section and you can slice it so.

X-ray has a method to slicing the data and then you can just plot the data so you can rather than actually you rather than subsetting at the opened app server, you can read in the whole file and then slice it as you need it. So subset it as you need it. The one great thing about this is.
Is X-ray does something called lazy loading. Lazy loading is basically where it does not care.
It doesn't load all the data set. It actually is not downloading the data, it's only downloading the data metadata and then what it does is it just subsets it and gets only that part of data.
That you actually need so X-ray has this functionality. So if you're not doing it on the opened up servers you can do it via this the best way about this is that you can then give specific latitude longitudes rather than the index of the latitude longitude. You don't have to say give me the 1st 100 or the 2nd.
100 latitudes along you can say given me from 60° N to 60° S or whatever it is. So that's a very good functionality issue. So I think I answered subjit's question.
But if you want to have follow up questions, please feel free to ask.
Yes, Tom, go ahead.

Tom Keel 51:10
Yeah, I was just going to say, although something like opened apps gives you this option, it's probably not. Oh, and sorry what is opened up? Yeah, it's just like a protocol that was designed by was it designed by NASA or something? I think.

Amulya Chevuturi 51:24
Yeah.
Yeah.

Tom Keel 51:30
For getting data from these, like massive climate models, so like each modelling group might return like terabytes of data and so that's all stored in one place and opened app is a protocol for getting access to that data which is stored on a server somewhere. But I would say.
When you do work with these models, for one thing, it's not great to just use one in isolation. So usually what's done climate scientists do is they log on to a

supercomputer like Jasmine and then take a subset from there. So although this is a way of doing it and testing things like.

Amulya Chevuturi 51:55
Mm hmm.

Tom Keel 52:08
Opened up if you were doing like a proper thing where you were looking at something across the climate models, you wouldn't use opened up. But I like Amelia. Obviously like this is just the example of like showing how you would get it if you were doing it this way, yeah.

Amulya Chevuturi 52:21
Yeah. And also it this is a way Tom adds a very good point. Usually we will log into supercomputers and directly do it. So Jasmine servers, which is UK's supercomputer, we have access to this data already there. We don't need to use these opened app servers but.
There are people around the globe who will not have access and then they will have to use open app service or other type of service. Open app are going out of fashion nowadays. They're not used as much and the next section is where we are actually going to discuss. How are the latest versions of like downloading data.
Any questions?
So the final section is reading multiple files together in opened app. I've put these as examples. I'm not going to go through the examples here, but you're more than welcome to try it on your own. So for now I'll just leave it for now. The next one of the most common ways nowadays to download data is.
APIs which are application programme interfaces, web pages now. Sorry data servers are now being hosted via this and there are many different examples of how this works. The most common example is the climate data store for ECMWF European. It's a European programme and they have like loads of data stored and usually you will have every data set that you want. You go to the download option and then you go down and then you have like I have to register, but I will get the API request.
So API request is basically you have to write a Python script to access the data. So there'll be some functions and the data will be accessed. So usually each web page will have their specific.

API request format written for you, so you should go to the documentation and see we are not going to currently work through the ECMWF or sorry, it's not ECMWF, it's a European Union, a climate data store, but they they are one of the most common data repositories.
That is available, but their documentation is very good, so please feel free to read it if you need any help in downloading data via this particular data set, let us know and we'll let you. We'll we can help you through it. Another one is the Cosmos SBI. So Cosmos data set is the fdri nowadays.
But it's the Ukch runs this, so there is a lot of information available, documentation on how to download Cosmos data. They have soil moisture, actual measuring points in different parts of UK and actually other parts of the world. Also where if you see this is their.
Network. This is how their network looks like and today what we are trying to do is we are going to try to download data using. So we are going to download one station data, the Alice Holt and we are going to download I think.
Daily temperature maximum which is ta Max. So we are going to download this particular data set for this station. The ultimate aim what we are trying to do here is we are going to get the station data then we are going to get ridded data from a different source and we are going to compare the observed data and the gridded model data to.
See how well they compare with each other, and that is the rest of the sessions objective. So we are going through some functions which are pre written for you. These functions they're not pre written by me, these are written.
By the Cosmos team and hosted on their website. So that's why reading the documentation is very important because you will get all the information that you need on how to download data in the documentation so you don't have to write it by yourself.
That's the great part. And then you'll have an easy way to download the data. So the first thing is to import the required packages. So if you run the cell then these are the pre written functions for the API web page package for the cosmos I would say just. Run this command and you can see that all of these functions are available in these examples that I've written in the comments section. So all of them are available. You don't have to worry about it, you just have to copy paste into your Python script or your Jupiter notebook. Whatever may be the case. So there are different ways to download the data.

So now.
Right.
So we are extracting station observation. So if we go to the next cell we are accessing station data from 2016 to 2022 start and end date for Alice one.
Station. That's the site name Alice Holt is the ID is Alice one. So if you click on this web page, the web page is given. Here you will find all the information that you need. You will have the ID of that station.
So basically you want to download the data for this and sorry yes, you want to download the data for Alice one and you want to download the DA Max data so the air temperature daily maximum air temperature.
You have to give all of this information for it to run. You have to give the base URL. So what is the base URL and the stite info. So ID locations. This all is given in the documentation. If not I have provided you this information.
And basically again you go through the different commands that are there in the documentation and.
If you run this you get.
You have given it the information that you need. Basically 2016 to 2022 Alice data. What variable you need and then you first get the metadata for the cosmos station. So this whole cell if you run it, you will get the metadata associated with Alice one which basically gives you the full name.
The latitude, longitude and many other information. What are the soil type so on and so forth. But specifically what we need from this is I need to know what the latitude and longitude for the data station is. Why do I need that? Because as I said, our objective is to link it to the grid.
The data set the model data set so I need the site latitude and longitude so that from the model data and extract the nearest possible station the location the grid point to this Alice site. So I need to extract the latitude longitude.
The next cell basically does that. It extracts the latitude and longitude for this particular site, so it gives me the site latitude and site longitude by getting this coordinate information. So from this data I can get the coordinates latitude and longitude and then post it.
In save it as the variable site latitude and site longitude. That is what this next cell is doing.
Yes, Ashlyn's question is, can you access multiple parameters of one? Did you if you mean parameters, do you mean variables Ashlyn?

Ashling Laffey 1:00:08
Hi. Yeah, I'm just from sorry. I'm just trying to find the cell. It was in the extracting the temperature Max parameter. Would you also be able to input in things like precipitation or whatever else might be?

Amulya Chevuturi 1:00:13
Yeah.
Mm hmm.
Mm hmm.
Yeah, yes, you can. If not, you can write a Python loop by itself and then multiple file like you can download multiple variables. So yes, it is possible. Yeah. And you can do multiple stations at a time and you can do. Actually you can do a lot of things.

Ashling Laffey 1:00:31
OK, lovely. Thank you.

Amulya Chevuturi 1:00:37
Cosmos API is really well written, so data downloading via API is as easy as the people that they make it. Cosmos API is very well written, I must say I've used it. All the functions are there so you can't it's it has a lot of flexibility. Sometimes APIs don't have that much of flexibility and you will have to download.
The whole data set and then subset it yourself, yes. Matt had post Matt has posted how to do it and you can figure it out. You can see it again. Examples are there on the web page so you can't follow. I'm just using one example as.
As just a way to showcase how this can be done. So in the next cell extracting cosmos data. So as you mentioned, you're basically downloading the station data using a pandas data frame. Again, pandas data frame is a Python package.
And it it's it's a way to download CSV type of file not download work with TIME series data, CSV type of files. So you'll get something like this. So basically you have the time you have the station ID and you have the temperature of that.
Day and you will have that for multiple stations. Sorry multiple time steps as you see there are about 200 and 2556 data sets.
Yes. Does anyone have any questions?
OK. If not then what we are doing is I'm taking this whole data frame and calculating

something called climatology. That's very common in in our hydrological sciences. Basically, climatology means you are calculating the mean for all the months or days. In a year, So what is the normal temperature maximum for each month, and that's what the cell is doing basically for 2016 to 2022 it is calculating date time, so this is January. This is the average temperature.
For January, for that particular station, over 2016 to 2022 and so on. Still the 12th month. So now we have the climatology or the average temperature month monthly average temperature for Alice one.
Station. So this is what we have done to get from API. This is an example that we have run. Now the next is we are going to take model data, get exactly a grid point nearest to the Alice one on the model data and then we are going to extract the model.
Data compare it against observations. This is one of the most common things that you would do in hydrological sciences or hydroclimate sciences. There'll be a model data. You'll compare it against observations to see what the biases are in your model or, and then you will go ahead next because models will have biases, there is no model without bias.
I will just stop here for a second before moving on to the next section.
Is anyone having problems you can run through the cells on your own time, but the basic thing is you need to understand if you understand the concept of API thing because most likely Cosmos API is not what you're. I mean, not everyone will be using Cosmos API.
Itself, but you should understand the basics of what an API is, how you would have to work with it just in case. If you have to have. If you have to use data from there basic summary you have to get you have to read the documentation well, understand the function.
Then just copy those functions and try to 1st play around and then get the data as you need. If you have any issues. Usually if you write to the support people they will get back to you into what errors you're making but the documentation is your greatest help in working with.
API data sets and API data sets are very common these days.
Brooke, sorry, we are not including bias correction in this particular session. I'm so sorry about that. But yeah, we don't have bias correction in the session.
And one more point. Thank you Kit for putting that about the restful APIs, but I have added although I'm going through this with you, I have added a lot more

information in the comments section.
Of the notebooks which you can go through it with your own time, there are also specific links that you can follow which will have more information that you can add to your.
Repertoire I would say OK.
Yes, Matt has added other notebooks which are different examples for using Cosmos API. So you have that too. Please feel free to use it, but again most of most of the API websites will give you exactly the script that you need to use to download your data. So you don't have to worry about it. It might take a while to understand, but you will get through it as long as you understand the basics that you can download data using specific packages or functions.
OK, now moving on to the not most newest way of downloading data sets. So cloud storage access. If you remember when we had gone to the NASA website, there was something called the Amazon Web Services or the simple storage.
Service the S3 version so this is what is a cloud storage. So nowadays all of these observation data sets. Oh, sorry all of the hydro climate data sets are slowly moving on to object store, even API sometimes are stored in the cloud and then API is just a way for you to access the data.
They are being stored on the cloud, but there are specific ways when they're stored on something called this S3. What I mentioned. Sorry, where did it go? The simple storage service when they're stored in that there are simple ways where you can access the data.
So these data sets are when they're stored on the cloud. They are quite similar. They're stored in the same format, but there are slight differences, so it's good to get to know these differences. So usually on a normal system we have a file system, then it goes to a.
Disc which is like has directories and then then there are the files, right? But the terminology is slightly different on a cloud storage it may be called an object store or a cloud store. Then instead of a disc you have something called a tenancy on which they're hosted.
And then instead of directories you have something called buckets, hilariously named, and then you have objects inside of files, and that's why it's called an object store. It is weird to remember, but as long as you remember that basically they translate into the same thing as the normal system, it's.
Easier to get to why they're very easy is they are very efficient and thus very cheap,

and that's why people are moving on to the cloud storage. Obviously let's not get into the detail of their climate impacts. They have a very severe carbon. Requirement because these servers are having a lot of like energy and water requirements. But that's a talk for another day. But these are the cheapest so lots of data data stores are moving to cloud storage.
So datas data is stored as objects rather and it scales out more than file systems, so it's easier to access, more efficient so on and so forth. The benefits are that you just access the data, you don't have to download the data, you can just access the data. Why are these HTTP links and you can manage them easily and they're very collaborative. So basically you have the data stored somewhere and then no one needs to have their own local.
Storage like you don't have to download it, you can just work off that one store, so you can when we mean collaborative, it can be internationally collaborative. So for example, we work on Jasmine which is AUK supercomputer, but getting people international partners to get access on that is quite difficult for us.
So instead of that we can host it on this Jasmine Object Store server, make the data open and then anyone can use it and our partners can use it. So it is very easy and the best part is that you can get like just subsets of data and work with it so.
People in countries who don't have a lot of like very big bandwidths can also work with this kind of data because as I said, these X-ray packages and all are very attuned to work with this.
Kind of data set and they just load the metadata. They don't load any data unless you actually finally decide to either analyse like you know, actually do any calculation on those specific data sets or actually plot it out. Then it will go and collect only that particular success data and just work on that.
Tiny bit so you can actually work very efficiently, and people who don't have a lot of bandwidth, network bandwidth can also work with these data sets. So it's it's very collaborative and it's very, very efficient. So there are different object stores, AWS, the Amazon Web Services.
And other other boat of three, and so on and so forth. I think now there is Google Cloud storage, there is Azure, Azure is I think Microsoft. So there are many different available. So it will depend on what kind of storage you have. But currently today for the demonstration we will work.
With the seed Jasmine, Jasmine Object Store so Jasmine is the UK supercomputer, we have an object store there. We have a tenancy which we are going to use this as an

example and that's where we have stored the data and made it Open Access so anyone can access it.
The one thing is that Matt Zalipiage and myself have actually written a whole GitHub repository which I have linked here, which gives you ways of like accessing object store data in much more detail. Again not going through it today.
But it is available. You are more than welcome to find it and work with it. And we have a whole video on YouTube on how we worked with this Jasmine object store. So you are more than welcome to go. Listen to it. I've linked it here. It will tell you about what the difference.
And ways of what are the differences between the object store and normals repositories? How did we upload data? How did we do conduct some experiments on it about like which is the fastest methods and so on and so forth. But I'm not going to go into that detail but it is here if you need it.
So now what we're going to do is on this seed adjustment Object Store we have we have data for the daily temperature maximum, but model data. So we have model data.
Stored there for 2016 to 2022, which we used it for the Alice one station. We are going to extract that particular data set for different ensemble members. Ensemble members are models run in different initialization, so you have multiple.
Realisations of the model we have 4 ensemble members here we're going to access all those force model data for a specific site closest to Alice one, and then we then we are going to present it as a plot. So basically this is a comparison method.
So again, we are importing the packages. These are very specific packages used for specifically the object store, the F spec, S3FS spec, ZAR. Again, I'll come to that what ZAR is? And X-ray is the package.
Python sorry, Python package which reads net CDF files, but it also reads ZAR files. Zar files are similar to net CDF files, but they're just storing data in a different format. When you read an X-ray file.
A ZAR file in X-ray. It's reading exactly like a net CDF file. There is no difference. The only different about ZAR files is and. Again, Matt will go into maybe some detail about this is that they are just more optimised for these cloud storages, so they have been created for these specific.
Specific reason to be stored on these cloud storages. Again, you don't have to worry about it because they work exactly like like CDF files. Unless you're creating these R files, you don't need to worry about how to. What is the difference between a net

CDF file and as R file?
Because when you read it on X-ray, it looks exactly the same as an STD file has the same latitude, longitude variable temperature. I mean sorry time so all similar. So we'll import these packages and then I've written this function.
But clearly to download the data from the storage. This is based on like it again documentation is available, but you're more than welcome to use this function in this function you would need something called the tenancy remember.
We had the tenancy here, so tenancy is basically the web link for the storage where you have stored the files. So you would need that and then you would need within that tenancy you would need the path to your file. So here we are using.
This ZAR file and this is the specific ZAR files that we want, which is basically the temperature, maximum ZAR file and this is the ensemble member number. So that's how we have named the file. So all you need to know is these two things.
And then you can use this particular command to basically read the file in.
So this sets the file name and then you just open it in the start file. So that's what this function does. I'll just run this function and then we will go on to the next step where we basically first before reading in the file, we'll explore the tenancy.
Um, I'll just take a quick break. I know we have 10 minutes left in the session, but I'll do you guys have any questions?
OK. I'll move on. So this is the Jasmine URL where we have stored the data. When you use your own data set you will get your own tenancy, but for now we're going to use this when we run this we have this.
Stored in a variable and then we can open this in this endpoint URL. Basically using this S3FS package we can explore what are the different files that are stored within. This path, so this is the path S3 is the as we mentioned is the simple storage service. So in this link within this path what is the different files that are stored. So when you run this command you'll see that there are different files.
That are stored so HURSHUSS these are, I think, specific humidity and relative humidity. Yes, and PR is the precipitation P surf is the pressure surface pressure and so on and so forth. So we need the T Max. So we need the T Max.
Because that's what we are comparing against Alice. So these are the different files. So now we know that we are using this particular.
File path. So this is a good way to find the file path if I changed it to Ensemble 4, there are four ensemble members. Unfortunately, they're named weirdly. It's ensemble member 1413 and 15 or something like that, so they're not following.

Yeah, they're not following.
This is very common again in climate data sets. You will have ensemble members which suddenly skip, so you have to be very sure. So again, go to the documentation, read the documentation. It's a good way to follow this. So if you read ensemble Member 4, then you have.
The other ensemble member data, so you have again an ensemble member data. If I do something like 5, I don't think that's five. I think it's the.
There is nothing because there is no ensemble #5 S going back to ensemble #4 and we have the different files stored there. So again this is a different way to explore your data tenancy. What is available within that tenancy.
Quickly now the next step is where we are going to actually access the data. So I had the function openzar from S3. If you see this was the function and here.
All you need to fill in is your Jasmine URL that you had saved before the tenancy, and then you need to give the pass right? So ensemble #1 temperature Max and when you run this.
It takes a while.
It will take a second or two, and that's where your file is opened. So now your net CD your is our file is opened, but it looks exactly like a net CDF file. So you have your Y coordinate, your X coordinate and your time coordinate. Now why Y&X instead of lat and long? Because this data set.
Is stored in a British National Grid which have xy coordinates and they have a 2D latitude and longitude format, so it's a slightly different projection. So you do have latitude and longitude, but they are in a 2 dimensional format. So again you have to be very sure about the types of files that you're working with.
So that's fine. So what we'll do is we'll convert this just data file and in the next step we are going to set the latitude and longitude as coordinates and now we have.
The latitude and longitude within the coordinate file rather than in the variable files.
So now we have an X latitude and longitude. Sorry latitude and longitude have coordinates. Why we need to do this is because we want to specifically search for a latitude and longitude within this data set.
To work with specific the specific Alice one site, so we have to extract that. So moving on here again, what further we are doing is basically I'm extracting the wearable task masks.
And slicing it for 2016 to 2022, because again, that's the specific data set I need it for and now I have the ensemble member one for the specific time period that I need.

So I have two 2500 and.
These many time steps and I have the latitude longitudes or the XY coordinates specifically set up for ensemble #1 in the next cell. Basically we are going through the different ensemble members. As I said, there's ensemble number 46 and 15 weirdly named.
But it is what it is and we are getting the same data for what we had got for chess ensemble, #1 chess. We are calling it chess because the data set is called chess and again links are posted and you're more than welcome to go explore the data set if you want to.
Yes. So it's going through the different ensemble members and will save the data accordingly.
And while this cell is running, I'll just, oh, actually the cell has run. The next one is what we are doing is we have another function where basically it takes the latitude and longitude of Alice one site which was site, underscore, latitude and site, underscore longitude and.
Within this gridded data, the chess data set of different ensemble members extracts the nearest grid point station, so that's what this function does and will run this and basically what it does is.
We are just in this cell. Basically what it does is it's finding the nearest grid point and extracting data over that grid point.
This is a temporary. This is a temporary file, so I'm just showing you as an example. It has a small file that we extract and then we extract the Y&X coordinate of the.
From the data set, so basically exactly extracting what are the indices which are the closest to the Alice one station. We get that and this is what is important for us. We need this to extract the data for the temperature and then we delete the temporary file.
And the next couple of steps are I'm going to go through them fast because I realise I've run over. Sorry, that was bad time keeping on my end, but basically I'm getting the date, time values and basically extracting the specific.
Point of that, sorry, this 140 and 497 for the model data. So the next few steps just do that for you. This particular cell is converting the temperature in the model. Data is given in.
Kelvin, whereas in the station data was in degree centigrade, so the units were different. So I'm just converting the data and getting Kelvin to degree centigrade and then I'm stacking all the ensemble members together.

And I'm just running all of this and I'm getting the model ensemble members climatology. So remember we calculated observation climatology so we have the model climatology and it'll go through. This will take some time because they're processing through.
And then we have the list of months and what we do is we ultimately plot model against observations. So that's what here we are going to do. While this is running through, I would want to ask if you guys had any questions not on the method.
3rd itself, like how are we doing the calculation, but if you have any questions on understanding object store and how we get data from it, that is something which we focus on but I'm happy to stay back after.
The session ends to discuss any other questions. If you had like on actually the method itself.
Any questions?
I'll give it another second. While the cells are running and I'll go through the cell information afterwards before while the cells are running and if there are no questions, we do have other methods which we haven't gone through.
But nowadays we also have, like catalogues. Even Eidc had catalogues, which was the data centre for Ukch. Ecwf has a catalogue. So all the data that is stored is generally catalogued and you can get like whatever is stored. It's a list basically you can get all the data that is stored.
Then the sensor observations are also stored online. You get can get them real time and again. I've posted some links Fdri project in the near future might have one hopefully and then Google Earth engine is another good way to actually process data real time and get.
Data global credit data sets and although we are not going to run a tutorial on that, there is a GitHub repository which has been developed within UKCH to run through for drought indicators that you can use or Google Earth engines themselves have very comprehensive tutorials.
That you can use. Sorry for taking this is taking a little bit too long, but if you guys have any questions please do raise it. Now I think this is stuck at this point where the extraction is taking a while but it it will go through it takes some time sometimes but it goes through.
But do you guys have any questions?
I'll give it another minute.
OK, while the notebook is running through, I will show you how the final product will

look like when your notebooks run through. This is how you'll get the final plot as. So basically the black line is the observed climatology for the Alice one.
And the blue spread is basically the spread of the different ensemble members of the different models. Sorry, different ensemble members for the Chess Cape model data so that we have.
Showcased it here. They're obviously over different months, so you can see that chess data is underestimating the maximum temperature for February, whereas it's overestimating it for the summer months of June, July, August.
We are calculating model mean, so basically that's the spread and the model mean is what is usually compared. But here as you can see that although observations for most months are falling within the ensemble member spread, there are some months where the.
Modelled or data is not capturing the outcome well. So especially for February as well As for September, the model is not being able to capture as same as the observations.
Of course, there are some issues that we can talk about that um.
Yes, Matt has a good suggestion. Sometimes you have to just stop and rerun the thing because of because collapse have an issue.
Yes, let me try that and sorry, going back to the original thing, this is a good this is a good way to have like the errors in the model, but you also have to consider that we have taken the nearest grid point. So sometimes the error might be stemming because we have not taken the current.
Grid point or it might be stemming from the fact that the model itself has biases, or it could be that we might not have.
It might not have, sorry. It might not have the whole region considered. So sometimes we take a regional average rather than just picking out a cell. So there are different ways to approach this thing. I will go through the different cells again, so you get the data for different ensemble members for the.
Different time period from 2016 to 2022 and then you calculate the climatology as we did in LS1 station and then you're just running the plotting software. So basically you get the.
Plotted output as I was showing you before. Brooke does model spread mean model uncertainty. It's so uncertainty is a very difficult. It's a very good question. Model uncertainty is generally between different models as far as I understand.
Ensemble members actually show you the internal variability, so currently when I'm

showing you the model spread, it's not the model spread. It's the. Sorry if I misspoke, it's the ensemble member spread and that shows the internal variability.
Of the of this.

Brook Legese 1:30:41
It.

Amulya Chevuturi 1:30:43
Sorry, yes, go ahead.
Brooke.
Brooke.

Brook Legese 1:30:52
No, it's OK.

Amulya Chevuturi 1:30:53
Yeah. OK. Sorry. I couldn't hear. Yeah. So it's it's showing you the internal variability. So the it's the different like the extent of the. So when I said ensemble members ensemble members are different realisations.

Brook Legese 1:30:54
Yeah.

Amulya Chevuturi 1:31:10
So just showing you how wide the internal variability of this temperature maximum can be within the climate or the historical climate from 2016 to 2022 and when the observations don't fall within that, that means your model.
Is not being able to capture that internal variability of that particular variable, which is the temperature maximum in this case.
OK, I will stop sharing now, but are there any other questions? Sorry for overrunning a little bit.
We'll wait for another minute.
Well, then it you are more than welcome to stay or leave. I'll be there for another 1520 minutes. So you're more than welcome to stop by and ask questions. Yes, Shagun, you had a question?

I saw a raised hand.
Oh, thank you for the very nice comments you guys are leaving in the chat, but that's really nice. But again, I'm here for another 1520 minutes post your questions in the chat or please feel to us now. We can also answer questions about like the methods which I didn't go into much detail.
But happy to do that or you guys can, I will now put it up to Sammy to close off the session if no more questions are there.

Samantha Rees 1:32:47
There there is one question which is what are possible reasons? Oh, it's OK. It's lots of thank yous and how brilliant you are and what are the possible reasons why the month of July is overestimated in the model?

SG **Shagun Garg** 1:32:49
Yeah.

Amulya Chevuturi 1:32:49
Oh, did I miss?

SG **Shagun Garg** 1:32:51
Yeah.

Amulya Chevuturi 1:32:53
Oh.

SG **Shagun Garg** 1:32:58
Yeah.

Amulya Chevuturi 1:32:58
So I actually wouldn't know this answer because I haven't personally worked with the chess data personally, so it could be a couple of things that maybe they're overestimating the temperature because the model might be so some models are warmer.

SG **Shagun Garg** 1:33:01

Yeah.
Yeah.
Yeah.

Amulya Chevuturi 1:33:17
More than normal and you know that biases exist, right? So this could be because of that, that this is a warmer than normal bias in the model. So I particularly don't know the answer to that. And that's the question that you guys have to answer when you start working with the model data.

Shagun Garg 1:33:17
Yeah.
Yeah.
Yeah.
Yeah.

Amulya Chevuturi 1:33:37
OK, sorry I couldn't answer that very helpfully. Shakun you had a question?

Shagun Garg 1:33:37
Yeah.
Yeah.
Hi, yes. So thank you for the amazing tutorial and it's very helpful to. For me personally, I don't work with hydras. You have work more with remote sensing. So it's very helpful to understand how easy it is to use the data. So but if I want to learn more about what?
Does model do? What are the word data sets are available and sort of that thing? Do you recommend some some sources or some like tutorials for that? Like where can I get the whole picture? What kind of data set is available?

Amulya Chevuturi 1:34:10
OK.
Right. So The thing is you will not have one stop shop for data sets. I mean there are certain places NASA servers will have a lot of data sets. The climate data store that I showed, they do have a lot of data sets.

Shagun Garg 1:34:14

Yes.

Yeah.

Yeah.

Yeah.

Amulya Chevuturi 1:34:28

But there is no one stop shop for any data. So with you for remote work remote sensing data. Although I'm not as expert on remote sensing data, a lot of the remote sensing data is stored on Google Earth engine, so it's good for you to explore that the links are posted. There's a.

Shagun Garg 1:34:31

Yeah.

Yeah.

Amulya Chevuturi 1:34:46

GitHub repository that is linked within our own GitHub repository. So if you look at Workshop 2 folder you will see that I have posted another GitHub repository which you can just follow on your own. So that's one way the best way. Google is the best way. You have to start Googling. You know what type of.

Shagun Garg 1:34:52

Yeah.

Yeah.

Amulya Chevuturi 1:35:06

Data sets you want to work with so specific data sets. This is a literature review that people will have to do on their own. Unfortunately, yes, you're right, there is no one stop shop for any data set, but you can find data sets quite easily with even a cursory.

Shagun Garg 1:35:17

Yeah.

Yeah.
Yeah.
Mm hmm.


Amulya Chevuturi 1:35:24
Look on the Google. So depending on the type of data set you need, you'll find it there, but you're more than welcome to come talk to us. Post an e-mail if you're having struggling to find any data. If we know we will let you know.

SG
Shagun Garg 1:35:27
Yeah.
Yeah.
Yeah.


Amulya Chevuturi 1:35:41
If you don't know, sorry, yes.

SG
Shagun Garg 1:35:41
Yeah.
Thanks a lot.


Amulya Chevuturi 1:35:46
Yeah. Thank you. And I will go to Brooke next before I follow, go for the chat questions. Go ahead, Brooke.

BL
Brook Legese 1:35:54
OK, Amalia, thank you so much. Again it it was a wonderful session pretty. I appreciate that. My question is regarding you showed us very well how to you know reach out those data how to download, extract.


Amulya Chevuturi 1:35:57
You.

BL
Brook Legese 1:36:11
From all what surprised me that you can compute thing cloudily without

downloading that one is very fascinating and I'll I'll check it out after a while. Yeah, yeah.

Amulya Chevuturi 1:36:19

Yes.

Yes, yes, absolutely. That is the biggest trend nowadays that you guys don't need to download any data. You can just do the processing. Though one thing it is I know you said it, this is semantics you are processing on your own computer, you're not processing on the cloud.

There are newer ways that are coming up, which I myself have not explored yet. Where you can do some of the processing in the cloud, but currently what we processed was on the Google servers because we were using the Google collab. So whatever processing you're doing.

Brook Legese 1:36:42

Thank you.

Amulya Chevuturi 1:37:00

It is happening on your local server so all your API is doing or API or whatever method it is. It is sending a link. It is sending a query to extract that particular data set.

That particular subset of data. It does come it doesn't get downloaded somewhere, but it is in your local RAM and then the process is happening on your local server. So you have to be very, very careful about the amount of data set that you're querying. Before processing, otherwise it will die here. I have tested it and I have kept examples where I've kept very small data sets, but if you just like ask for like a whole huge model data set your processing will not happen if you don't have the processing power. So that's something you should be very careful of and that was a very good. Question to ask did you have? Yes.

Brook Legese 1:37:54

How to check it out? Yeah. My question is actually related to downscaling. Now we download the datas. So how to downscale? I don't know whether it is within the scope of this training or not.

Amulya Chevuturi 1:37:58
Yeah.
Mm hmm.
Hmm hmm.

Brook Legese 1:38:10
But after that, yeah, how we are going to downscale it could be statistical downscaling metre actually dynamic is very tough in the. So is there any any tutorial that you will suggest us or any link that will lead us to?

Amulya Chevuturi 1:38:10
It's not, yeah.
Yeah, yeah.
Yeah.
Yeah.

Brook Legese 1:38:28
Yeah.

Amulya Chevuturi 1:38:29
I mean, I'll leave it to the other facilitators. I don't know any particular downscaling methods which are like tutorials that are available. I can post some links in the chat where there is some statistical downscaling that is possible.

Tom Keel 1:38:40
And.

Amulya Chevuturi 1:38:46
But that, as you said, statistical downs. Sorry, downscaling itself is such a wide you can't have like one stop shop. There is different downscaling methods, dynamical methods, statistical method. There are so many different methods.

Brook Legese 1:38:47
OK.

Tom Keel 1:38:54
Umm.

Amulya Chevuturi 1:39:02
So there is no one tutorial I know, So what I would suggest is if there is a method that you need to follow, you'll have to do literature review, get the papers on that and then create your own downscaling methods. There are some very easy statistical downscaling methods.
That you can follow easily, but yeah, there is no tutorial that I know of. Particularly. Yeah, sorry.

Brook Legese 1:39:30
Thanks so much. I appreciate. Thank you.

Tom Keel 1:39:30
And.

Amulya Chevuturi 1:39:32
Did anyone yeah, go ahead, Tom. Yeah.

Tom Keel 1:39:33
Just just add to that really quickly. Yeah, whichever method you choose, if you're familiar or, well, comfortable with X-ray, it'd probably be through that that people normally do this sort of stuff. So like the resample method. But then, yeah, like, as Amelia said, there's lots of different ways of doing it, but like.

Amulya Chevuturi 1:39:44
Yeah.
Yeah.
Yeah.

Tom Keel 1:39:53
Doing it via X-ray is a good way to do it, because you've already got the data in the format and then you can just increase. Yeah, increase the resolution, yeah.

Amulya Chevuturi 1:39:58

Hmm.

Yeah, yeah. I mean, we won't go into details of that, but that's like in itself a whole new topic. I think we'll have to run a one week's workshop on that. OK? One more question was there was some discussion going on about overestimating in the month of July, another reason why that may be.

Why the temperature is overestimated or underestimated could be that the precipitation errors are pretty significant in the model, so that might be changing the temperature that might be causing the temperature biases or vice versa one can cause the other.

Hmm.

Hey.

Yes, Matt, thank you for that. About the date list of places to start, look for hydrological data. That's great. We can provide Sammy some of the links, but a lot of the links are also posted in my GitHub repository. So go check those out.

Then Diane's question is observed, climatology does not necessarily match simulation due to the latter one represent aerial averages. Am I right? So particularly in the example that we have run, it is not an aerial average, it is a specific grid point. Which is closest to the Alice one latitude and longitude and that's another reason why we might be getting errors. So hope that makes sense.

Aerial averages actually would have been a better choice, but because again, processing issue, I didn't want to download like a lot of process a few grid points. So that's another reason and Matt also.

And that could you suggest a method for analysing the biases in daily precipitation data? OK.

The easiest way to start off with estimating bias is first just calculate bias, which is model minus observations and then you have many other ways to calculate, like there are many other. I want to skill scores. There are performance metrics so you can use person.

And there are many other ones I can again post the links in the chat or and then Sammy can have that there is a whole there's a link that I can give you which is like it's to create calculate performance metrics as well.

Skill scores of any model, and it's a very comprehensive WM OS web page link. I'll post a link Sammy later. OK. Any other questions?

The.
Oh, OK. Did I miss any other question guys?

Matt Dalle Piagge 1:43:05
I don't think so. Nice stuff.

Amulya Chevuturi 1:43:07
OK, if I've missed any of your questions, please post it in the chat or raise or no post it in or just just unmute yourself and.
Get.
OK.
Umm.
Yes. Could you remember I meant learning Julia for data analysis. I don't use Julia, so I can't recommend it to people. Can anyone?

Matt Dalle Piagge 1:43:37
Do any of it? Yeah, I don't think so far I've found Python is fine. It probably I think it does have some advantages, Julia, but I've not. Haven't put it basically this way. I haven't had the time to learn it and and try it out and see.
So yeah, I can't make a recommendation, I don't think.

Amulya Chevuturi 1:43:55
Yeah, I thought. I thought Julia was an operating system. Is it like, also, like, a language?

Matt Dalle Piagge 1:44:04
I thought I thought so, or maybe I'm wrong as well, but there's also rust these days, yeah?

Kit Macleod 1:44:07
It's definitely a language. Yeah, I mean, it is quite widely used in some Earth modelling communities. I mean, one of the reasons why people quite like it is one of the challenges with sort of Python And R is this sort of two language issues that you use sort of Python on the surface. But often, you know all the.

Amulya Chevuturi 1:44:07
Yeah.
OK.

Dayan Renán Saynes Puma 1:44:09
Another running language.

Kit Macleod 1:44:26
Computations being done in sort of C or below, whereas if Julia doesn't have those issues but it you know it is a smaller community, they have great resources online. You know some really great range of packages and you know annual meetings so.

Amulya Chevuturi 1:44:46
Sorry, go on out there. OK, I'm posting eleanora's question answer here. Sorry someone was saying.

Dayan Renán Saynes Puma 1:44:50
You.

Amulya Chevuturi 1:45:04
Matt is saying then that basically Python might be faster because there are lots of supporting libraries. Yes, Python has very good open source like you can get a lot of information about path and packages online.
So maybe the support is better? I don't know. Yeah. Sorry, Matt. Go ahead.

Matthew Fry 1:45:22
Yeah.
I'm saying that Julius probably fast people. You choose it for, for its speed, for analysis as well as that consistency that Kit mentioned. But I think Python is well supported with all the different open source and packages it's got for lots of different things. So it's a good.
It's a good, good and accessible, but that way, but if you're going to deep dive deeply and then maybe Julia. Yeah.

Tom Keel 1:45:49

Just just add to that like Python is the most popular programming language. Julia's probably not even top 10, so just you're not going to get much support if you try to do other things with it, so you save yourself a lot of time by learning Python because it lets you do things from web development to data analysis.
Julia would be very much a statistical language just for doing data analysis, and then you might not have support if you're just like for example. I don't know if there's a downscaling package and I've just found two from a quick Google, so things like that, the niche stuff you'll probably run into more problems. So if you're deciding between.
In the two, I'd definitely say stick with Python And the most popular and the most support.

Amulya Chevuturi 1:46:45

If there are no other questions, we'll let you go off for lunch. Sorry for overrunning, but yeah. See you at 1:00 in 40 minutes for maths.

Matt Dalle Piagge 1:46:58

Yep.

Amulya Chevuturi 1:47:00

Session.

HL

Hywel Lloyd 1:47:01

Thank you very much.

Amulya Chevuturi 1:47:04

And all the facilitators except I will see you at lunch place. Bye guys.

CB

Chris Bean 1:47:04

Thank you.

Samantha Rees 1:47:10

Bye everyone.

◉ **Samantha Rees** stopped transcription