Course Overview & Introduction to Python Programming

Chaitanya K S
Deputy Director
National Water Academy
Central Water Commission, Pune

Python for Data Analysis & WRM

- Clear understanding of hydrological processes, assessment and monitoring of various components of hydrological cycle are essential for sustainable development and management of water resources.
- This has assumed greater importance in the context of growing sectoral water demands and climate change predictions.
- Quantification of hydrological parameters through traditional methods of measurement as well as retrieval from remote sensing imagery will yield humongous amounts of data.
- Long term data sets at various spatial and temporal scales serve as inputs to process based hydrological models for better estimation and forecasting of hydrological parameters.
- Data preparation, manipulation, analysis, visualization/presentation are very important for decision making.
- Python programming with support of active user community and libraries has become a popular choice for data analysis tasks.

Python for Data Analysis (Contd.)

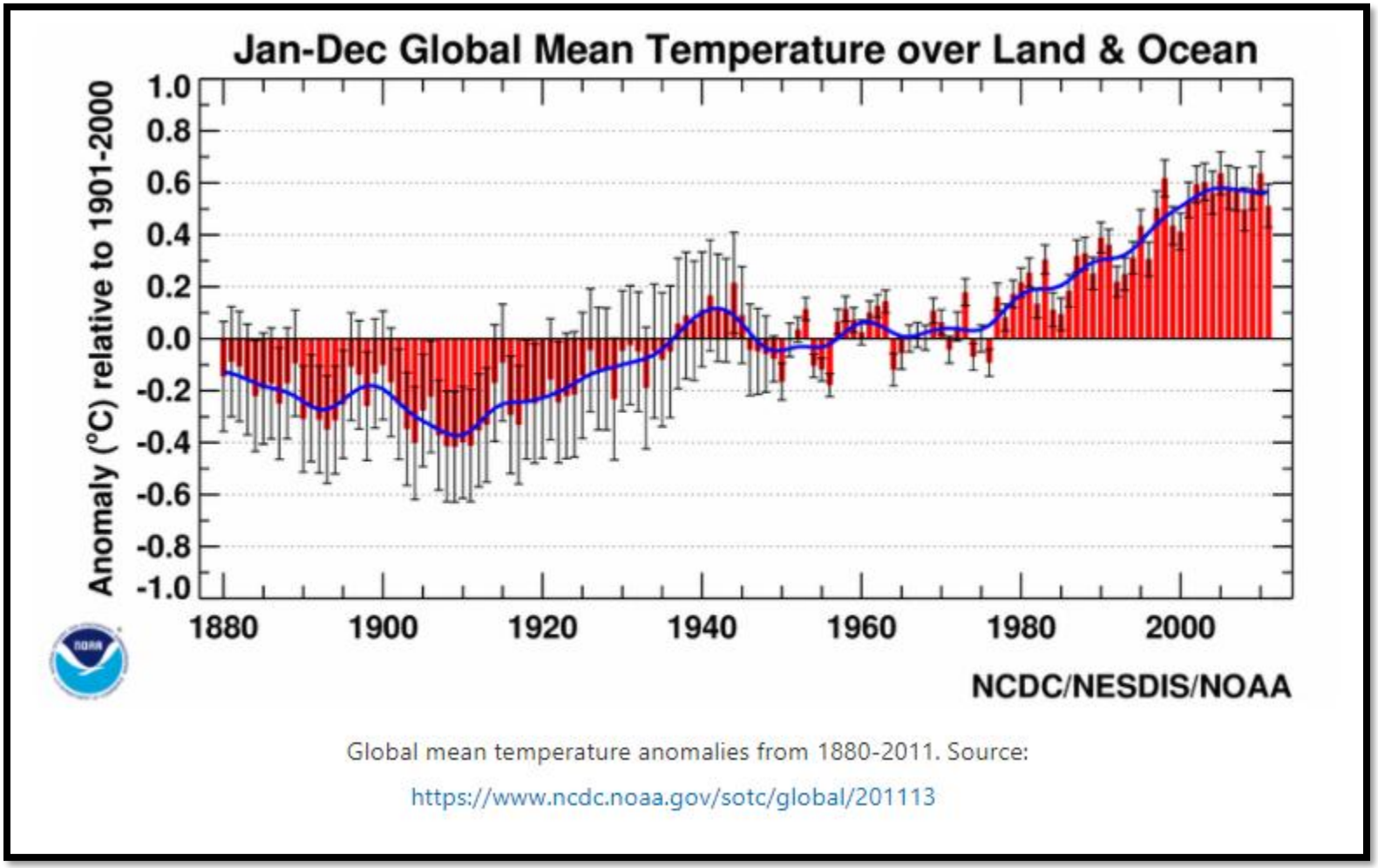
- Raw data often fails to convey the necessary information

USAF	WBAN	YR--MODAHRMN	DIR	SPD	GUS	CLG	SKC	L	M	H	VSB	MW	MW	MW	MW	AW	AW	AW	AW	W	TEMP	DEWP	SLP	ALT
029740	99999	195201010000	200	23	***	15	OVC	7	2	*	5.0	63	**	**	**	**	**	**	**	6	36	32	989.2	*****
029740	99999	195201010600	220	18	***	8	OVC	7	2	*	2.2	63	**	**	**	**	**	**	**	6	37	37	985.9	*****
029740	99999	195201011200	220	21	***	5	OVC	7	*	*	3.8	59	**	**	**	**	**	**	**	5	39	36	988.1	*****
029740	99999	195201011800	250	16	***	722	CLR	0	0	0	12.5	02	**	**	**	**	**	**	**	5	36	27	991.9	*****
029740	99999	195201020000	220	7	***	722	CLR	0	0	0	12.5	02	**	**	**	**	**	**	**	0	36	28	995.8	*****
029740	99999	195201020600	220	16	***	15	OVC	5	*	*	9.4	02	**	**	**	**	**	**	**	1	36	34	997.1	*****
029740	99999	195201021200	110	14	***	8	OVC	5	*	*	12.5	70	**	**	**	**	**	**	**	2	34	28	993.1	*****
029740	99999	195201021800	160	14	***	8	OVC	7	*	*	1.2	73	**	**	**	**	**	**	**	7	34	32	985.9	*****
029740	99999	195201030000	180	18	***	15	OVC	5	*	*	3.8	26	**	**	**	**	**	**	**	7	36	28	985.4	*****
029740	99999	195201030600	200	14	***	15	BKN	5	*	*	5.0	02	**	**	**	**	**	**	**	7	36	32	985.2	*****
029740	99999	195201031200	250	11	***	15	OVC	5	1	*	9.4	02	**	**	**	**	**	**	**	7	34	30	986.8	*****
029740	99999	195201031800	340	9	***	5	BKN	5	7	*	5.6	03	**	**	**	**	**	**	**	2	28	27	988.8	*****
029740	99999	195201040000	230	7	***	722	SCT	0	0	5	9.4	02	**	**	**	**	**	**	**	1	28	27	994.4	*****
029740	99999	195201040600	270	11	***	15	OVC	7	*	*	5.0	70	**	**	**	**	**	**	**	2	30	27	998.6	*****
029740	99999	195201041200	250	15	***	15	OVC	5	*	*	5.0	71	**	**	**	**	**	**	**	7	30	27	1002.8	*****
029740	99999	195201041800	260	7	***	40	OVC	5	*	*	12.5	02	**	**	**	**	**	**	**	7	30	27	1006.6	*****
029740	99999	195201050000	***	0	***	***	BKN	5	*	*	12.5	01	**	**	**	**	**	**	**	2	28	27	1009.8	*****
029740	99999	195201050600	***	0	***	98	OVC	7	2	*	1.9	71	**	**	**	**	**	**	**	7	27	27	1012.2	*****

Source: https://geo-python-site.readthedocs.io/en/latest/lessons/L1/motivation.html

Python for Data Analysis (Contd.)

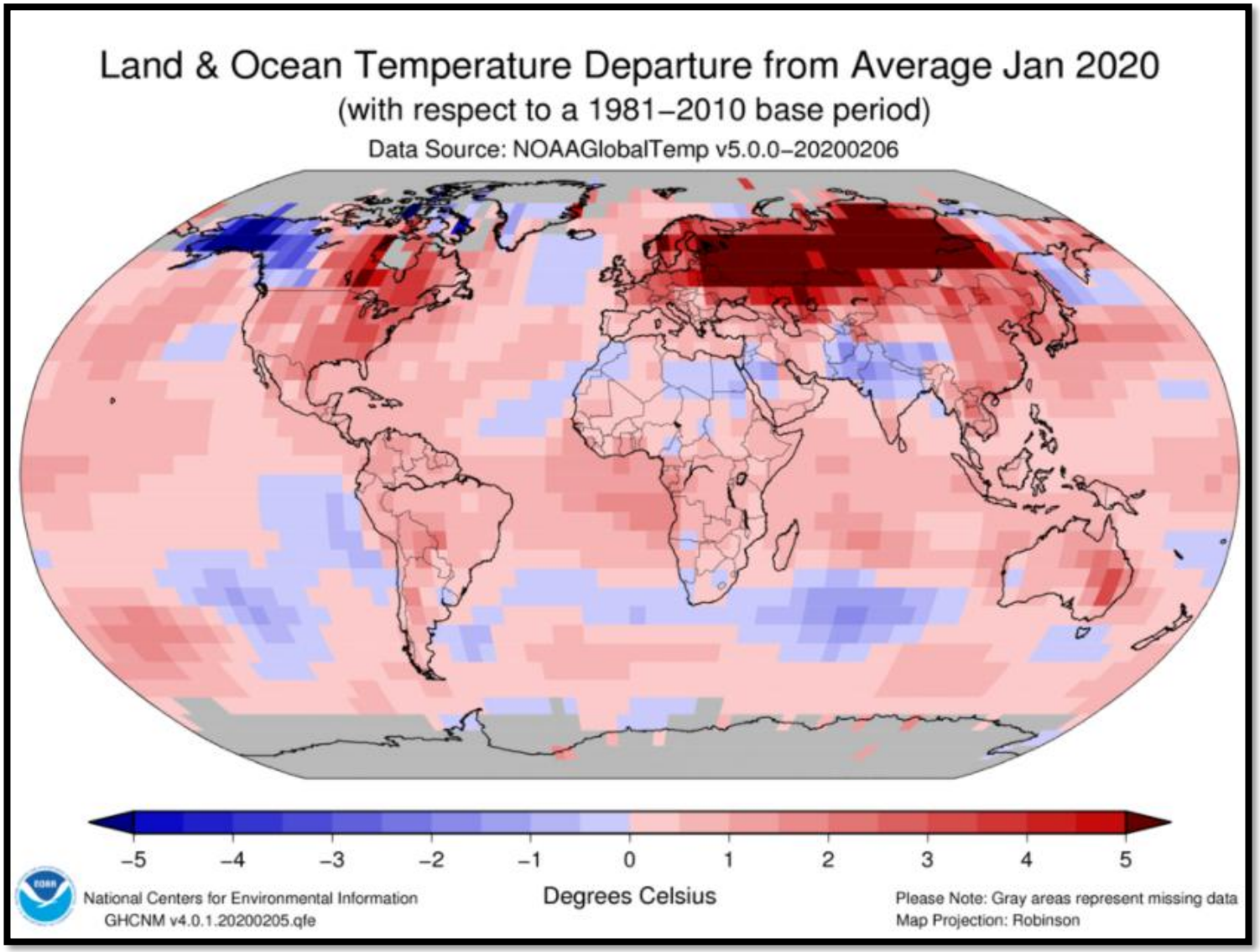
- Same data presented as time series plot



Source: https://geo-python-site.readthedocs.io/en/latest/lessons/L1/motivation.html

Python for Data Analysis (Contd.)

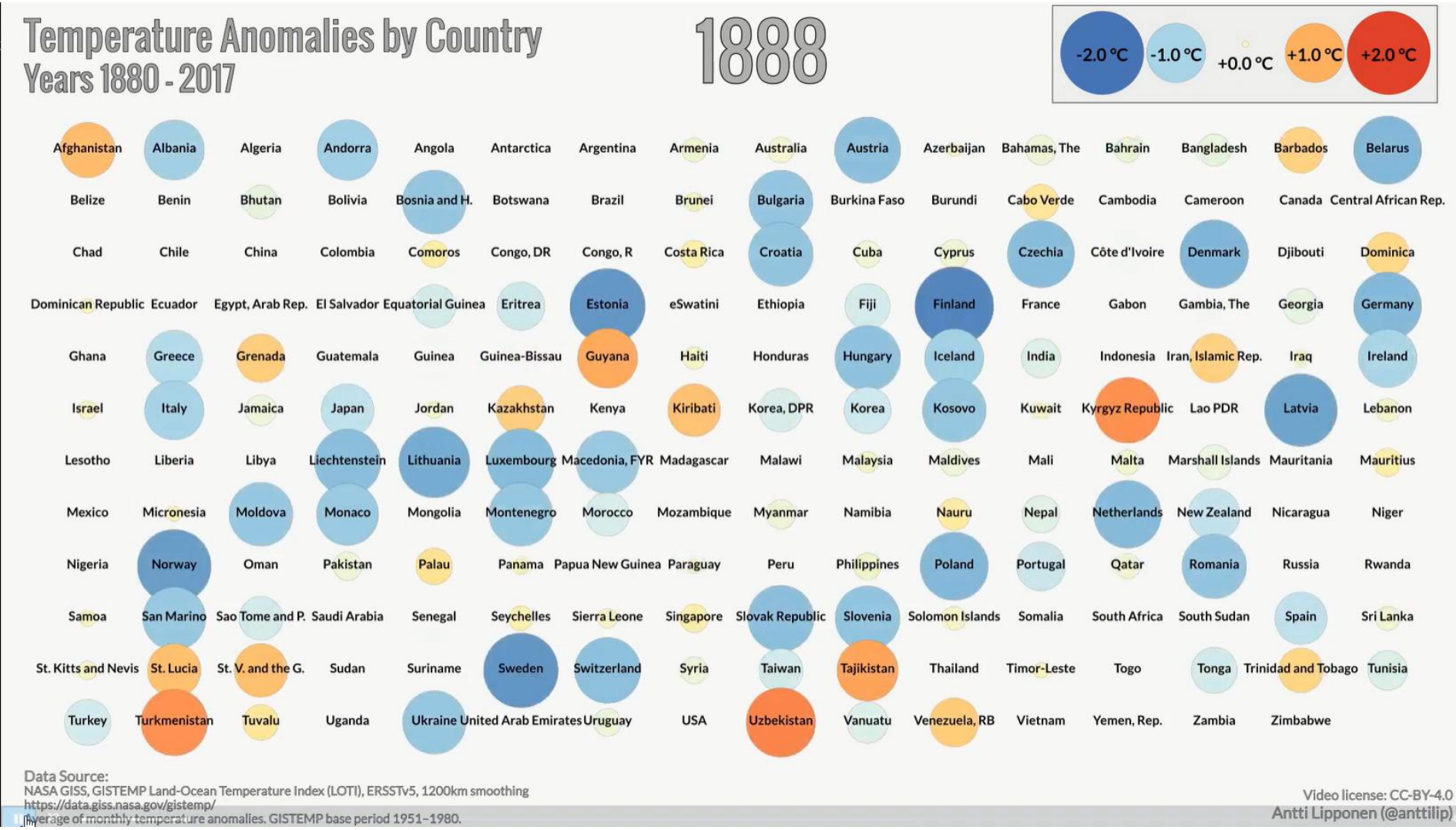
- Data with added spatial context



Source: https://geo-python-site.readthedocs.io/en/latest/lessons/L1/motivation.html

Python for Data Analysis (Contd.)

- Data visualization across space and time



(Source: https://geo-python-site.readthedocs.io/en/latest/lessons/L1/motivation.html

Geospatial analysis and programming

- Our planet is undergoing environmental changes that have never been seen in recorded history. Parallelly, open access data and software have become much more widely available to everyone.

- Geospatial analysis is the best approach to understanding our world more efficiently and deeply. It can answer critical questions at the heart of the choice to be made and often cannot be answered any other way.

- Why use a programming language when modern geospatial analysis can be conducted with the click of a button in any of the easy-to-use commercial or open-source GIS softwares?
  - You want complete control of the underlying algorithms, data, and execution
  - You want to automate specific, repetitive analysis tasks with minimal overhead from a large, multipurpose geospatial framework
  - You want to create a program that's easy to share
  - You want to learn geospatial analysis beyond pushing buttons in software

(Source: Learning Geospatial Analysis with Python, Joel Lawhead, Packt publishing)

Why Python?

Python was created in the early 1990s but has become more popular lately

Notable Features of Python

- Full spectrum language – Basic as well as advanced features
- Good for beginners - elegant syntax - easier to read and write
- An Interpreted language that can be picked up and get your program working without getting lost in problems of compilation
- Interactive mode makes it easy to test short snippets of code
- Comes with a large library of modules that can be used to do all sorts of tasks
- Easily extendable by adding new modules implemented in a compiled language
- Can also be embedded into an application to provide a programmable interface
- Portable - Runs anywhere – Windows, Linux, Mac and Unix
- Free and Open Source - It doesn't cost anything to download or use Python, or to include it in your application. It can also be freely modified and re-distributed.
- Active user community support for addressing bugs and other issues

Why Python? (Contd.)

C

```c
#include <stdio.h>
int main() {
   // printf() displays the string inside quotation
   printf("Hello, World!");
   return 0;
}
```

C++

```cpp
#include <iostream>

int main() {
    std::cout << "Hello World!";
    return 0;
}
```

Python

```python
# This program prints Hello, world!

print('Hello, world!')
```

Why Python? (Contd.)

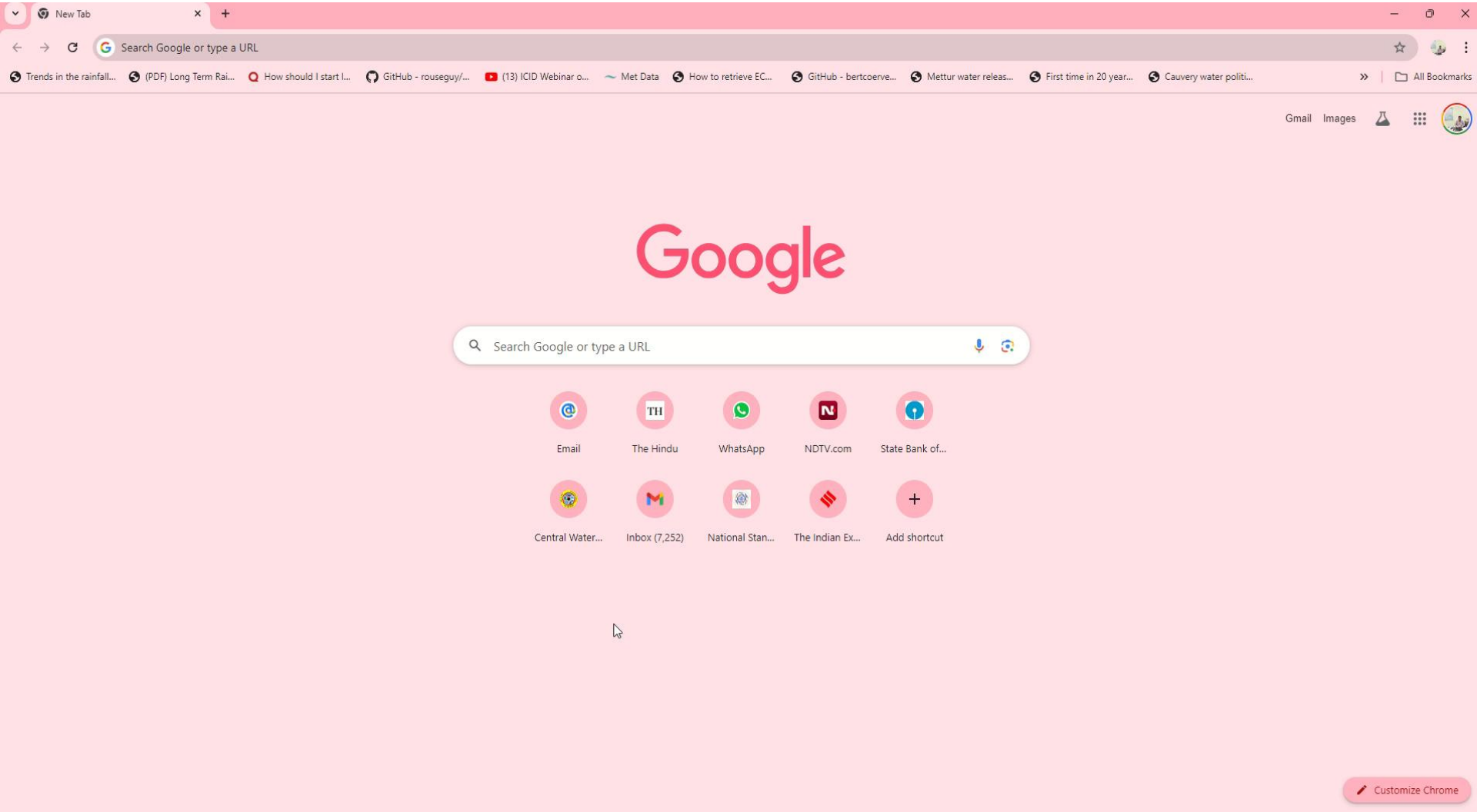
Some programming-language features of Python :

- A variety of basic data types are available
- Python supports object-oriented programming with classes and multiple inheritances.
- Code can be grouped into modules and packages.
- Cleaner error handling
- Data types are strongly and dynamically typed.
- Mixing incompatible types causes an exception to be raised, so errors are caught sooner.
- Python contains advanced programming features such as generators and list comprehensions.
- Python's automatic memory management frees you from having to manually allocate and free memory in your code

(Source: https://wiki.python.org/moin/BeginnersGuide/Overview)

Installation of Python

- You may already have Python installed on your system through Operating System (other than Windows) or through GIS software
- Online Python Interpreter: https://www.python.org/shell/



Anaconda Distribution

- What is Anaconda Distribution? : Anaconda Distribution is an easy-to-install collection of high-performance Python libraries along with Conda, which is an open-source package and environment management system.

- Conda finds, installs, runs and updates packages and their dependencies.

- If a package requires a different version of Python, Conda can set up a separate environment to run that different version of Python, while continuing to run current version of Python in your normal environment.

- Why do I need Anaconda Distribution? : Many scientific packages require a specific version of Python to run. It's difficult to keep various Python installations on one computer from interacting and breaking, and harder to keep them up-to-date. Anaconda Distribution makes management of multiple Python versions on one computer easier, and provides a large collection of highly optimized, commonly used data science libraries to get you started faster.

Anaconda Starter Guide

Managing Environments and Channels

- What is an environment? : A folder or directory that contains a specific collection of conda packages and their dependencies, so they can be maintained and run separately without interference from each other.

- You can have multiple versions of Python (and 3rd party packages) installed on your system within isolated environments which will avoid conflicts. It is a good practice to create separate environment for any new python project.

- What is a channel? Conda channels are the locations where packages are stored. They serve as the base for hosting and managing packages. Conda packages are downloaded from remote channels, which are URLs to directories containing conda packages.

Different channels can have the same package, so conda must handle these channel collisions. Good practice to install packages from the same channel to the extent possible.

Conda User Guide

conda-forge community channel

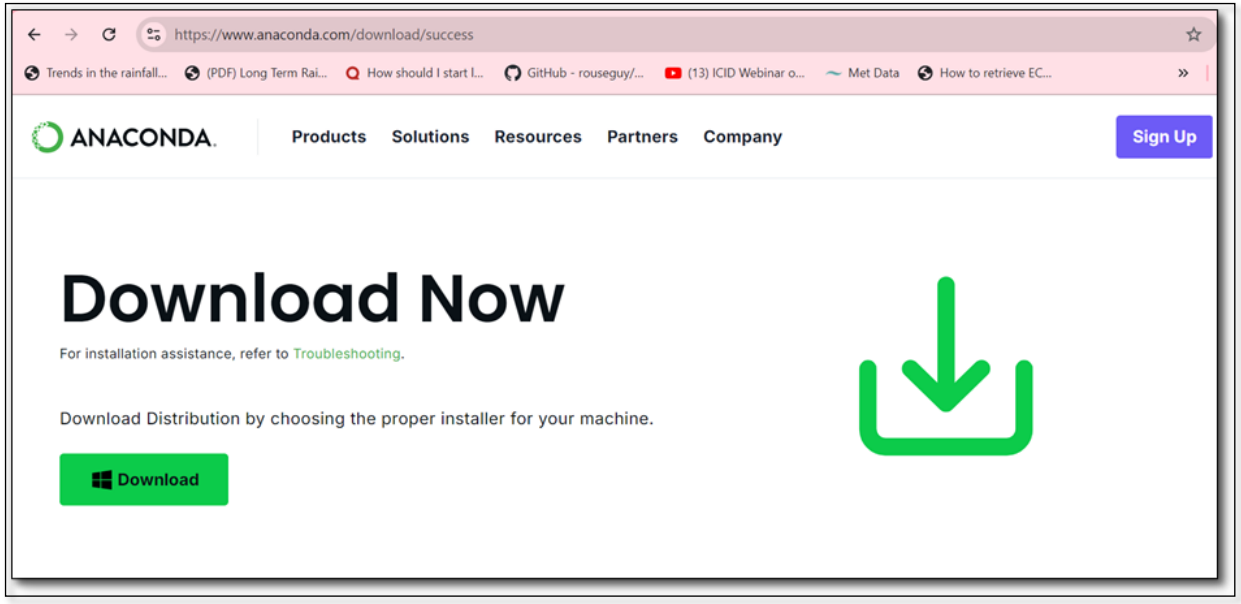
- conda-forge is a community channel made up of thousands of contributors that provides conda packages for a wide range of software.
- Before conda-forge locating packages was difficult due to them being scattered over many channels, Combining packages across channels was not always possible due to binary incompatibilities, Packages were only available for architectures the developer was interested in or had access to, channels were often abandoned, updating required locating new channels.
- With conda-forge all packages are shared in a single channel, care is taken that all packages are up-to-date, common standards ensure that all packages have compatible versions.
- If a package is not available from conda, you may be able to find and install the package via conda-forge or with another package manager like pip. It is not advisable to mix conda and pip for installation of packages unless strictly necessary.
- For this course we will install Python as part of Anaconda package manager and then create a new environment 'python-wrm' for installing various packages from conda-forge channel

Conda-forge documentation

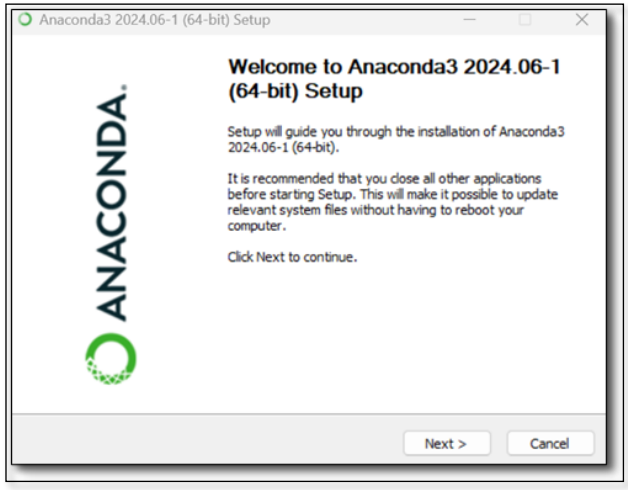
Installation of Anaconda Distribution

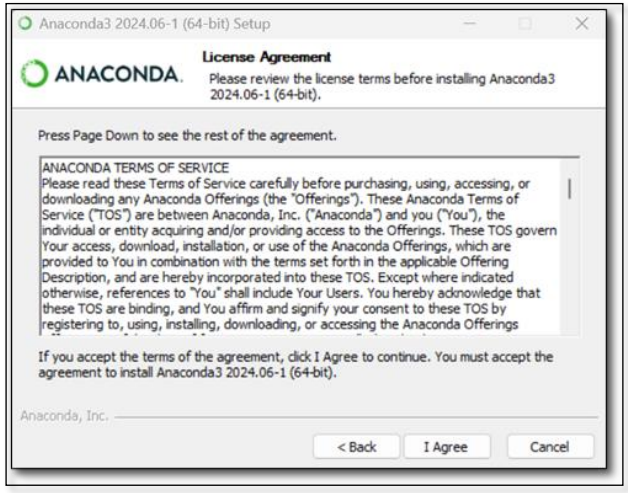
[Anaconda installer for windows]

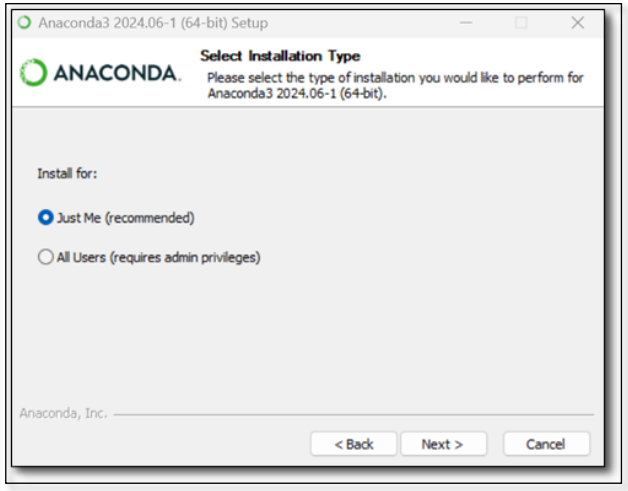
[Miniconda installer for windows]



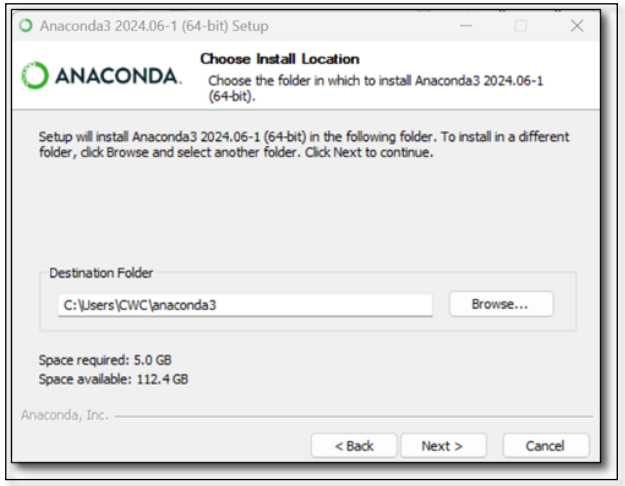
After downloading double click the installer to launch.

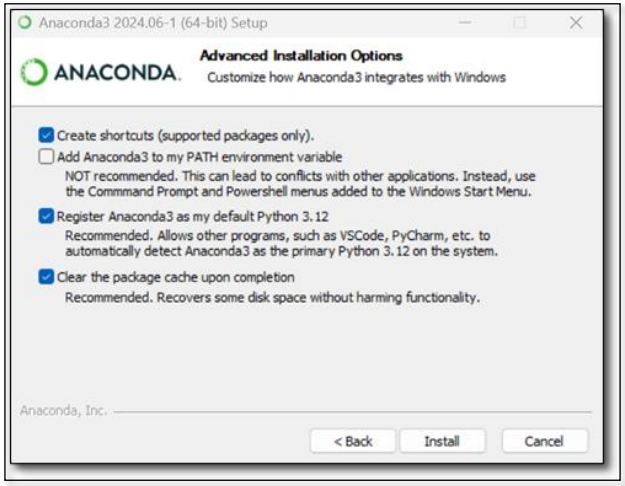


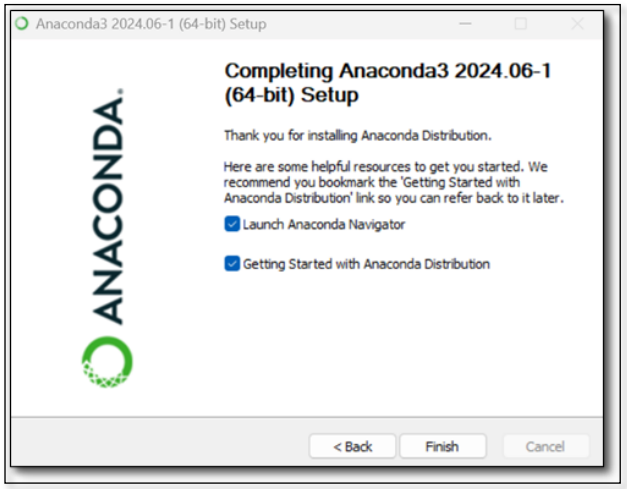




Installation of Anaconda Distribution (contd.)







Installation of Python Packages using conda

creation of new environment

conda create --name (NEWENVIRONMENT)

creation of new environment 'python-wrm' with specific version of python

conda create –n python-wrm python=3.12

listing of environments

conda info --envs

activation of new environment

conda activate python-wrm

installation of packages in the new environment from conda-forge channel

conda install –c conda-forge (PACKAGENAME)

numpy, pandas, shapely, geopandas, jupyter, jupyterlab, matplotlib, geojson, pysal, mapclassify, rasterio, geopy, pyrosm, osmnx, contextily, folium, mplleaflet, bokeh, descartes, rtree, rasterstats, richdem,scipy,pip

Thank you