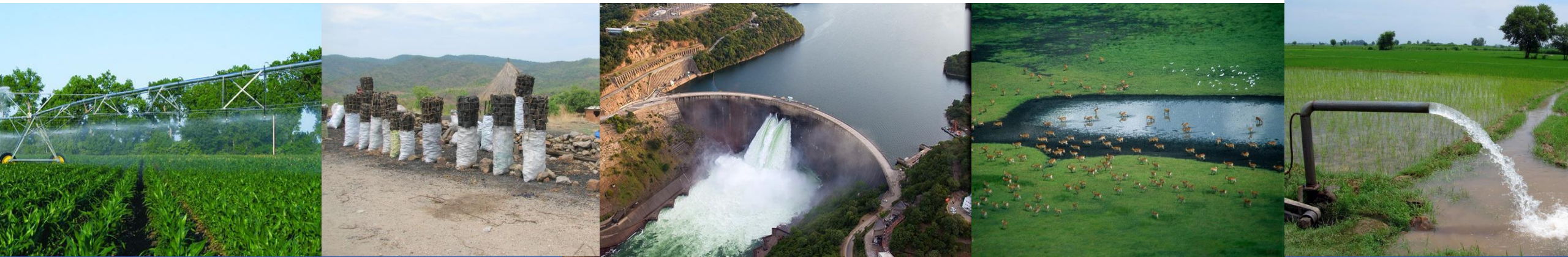


Exercise 6: Water cycles and signatures

Peter Burek, Mikhail Smilovic, Luca Guillaumot
International Institute for Applied Systems Analysis
Research Scholars at
Water Program



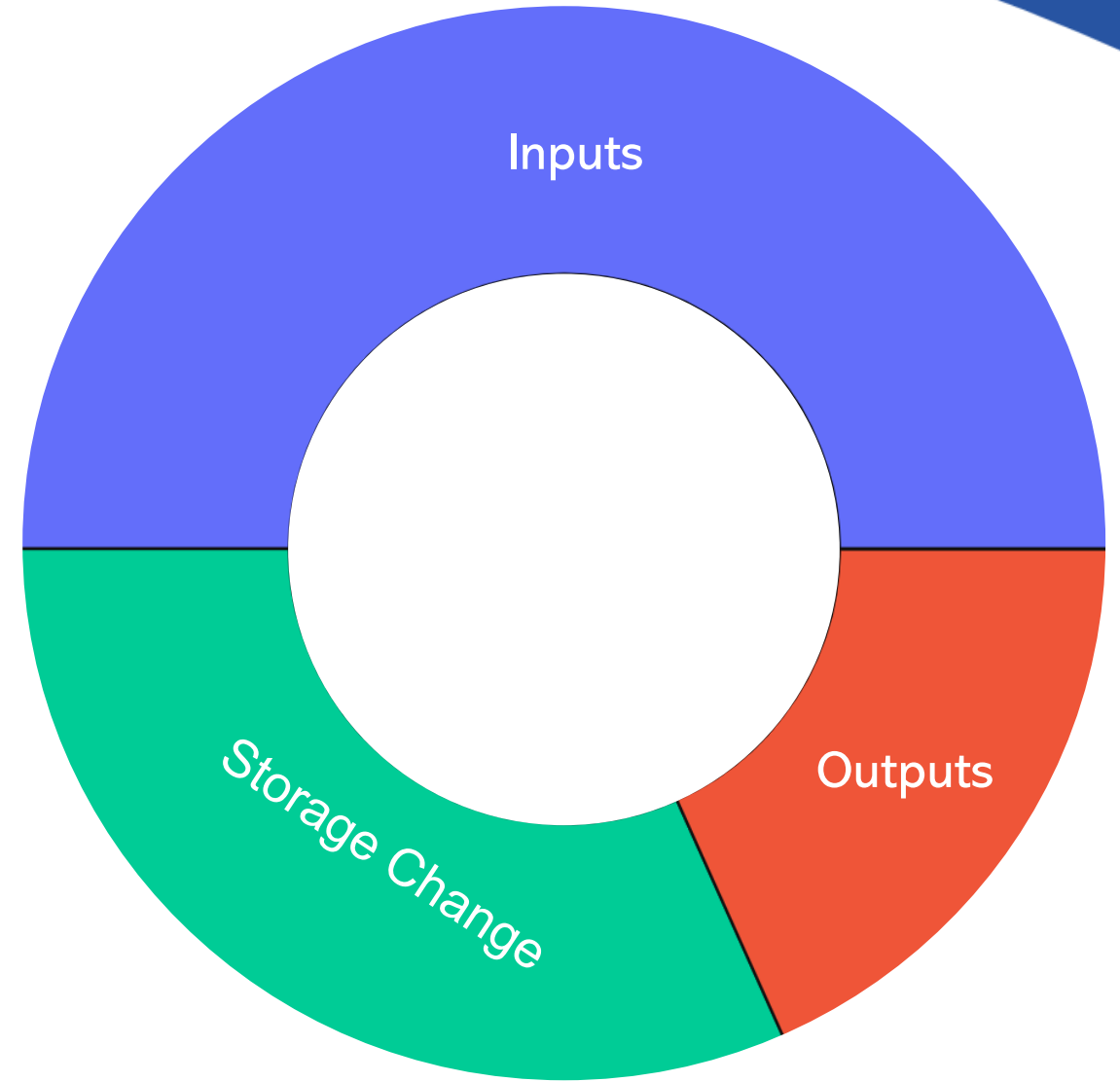
Water cycle

A water balance is used to understand the flows through a system.

A water cycle/wheel/circle summarises the flows through a system over a certain period.

The **inputs** to the system equal the **outputs** from the system plus whatever **changed in the system**.

$$\text{Inputs} = \text{Outputs} + \text{Storage change}$$



Water cycle and signature example

The Rhine basin for an example year



Water cycles and signatures

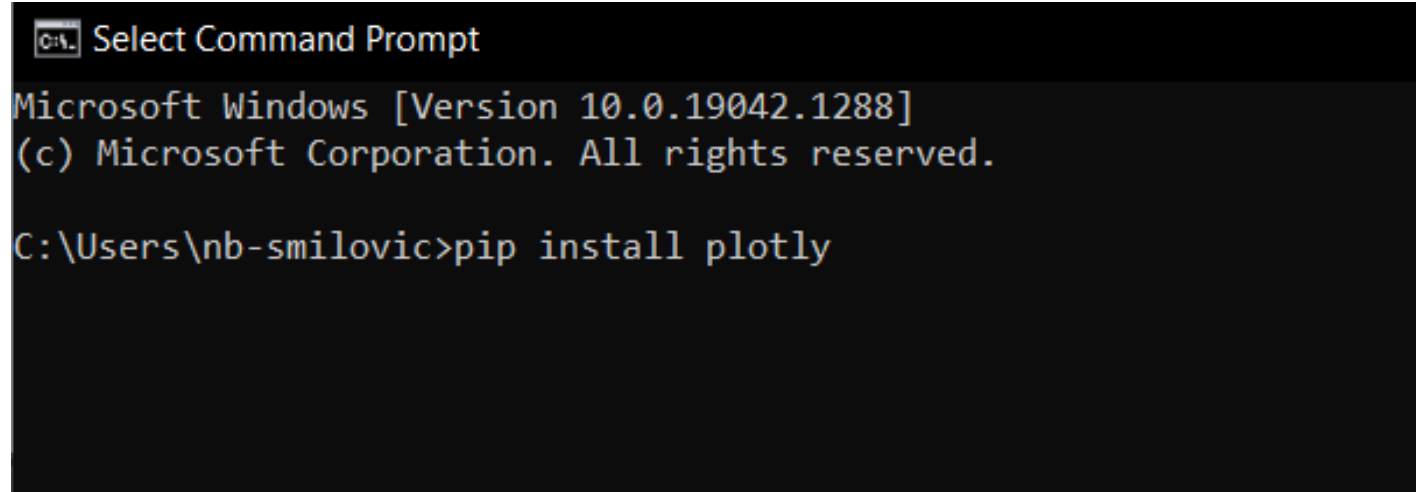
1. Install Jupyter notebooks and some python packages
2. Open and execute WaterCycles.ipynb using Jupyter notebook
3. Play CWatM with specific outputs and for your coordinates
4. Execute WaterCycles.ipynb, updating the outputs path and basin outlet coordiantes



1. Install four python packages

In a command prompt terminal, type the following commands followed by the enter button (or however you may be familiar with installing packages)

- `pip install plotly [enter]`
- `pip install pandas [enter]`
- `pip install matplotlib [enter]`
- `pip install notebook [enter]`



```

C:\> Select Command Prompt

Microsoft Windows [Version 10.0.19042.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nb-smilovic>pip install plotly
  
```

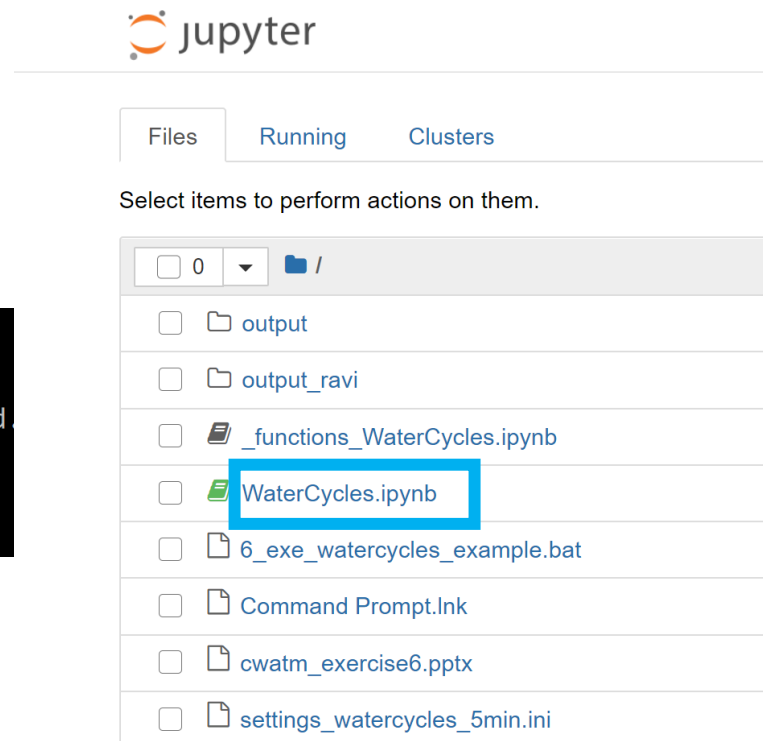
2. WaterCycles.ipynb

1. Open up the terminal within the Exercise 6 folder and type

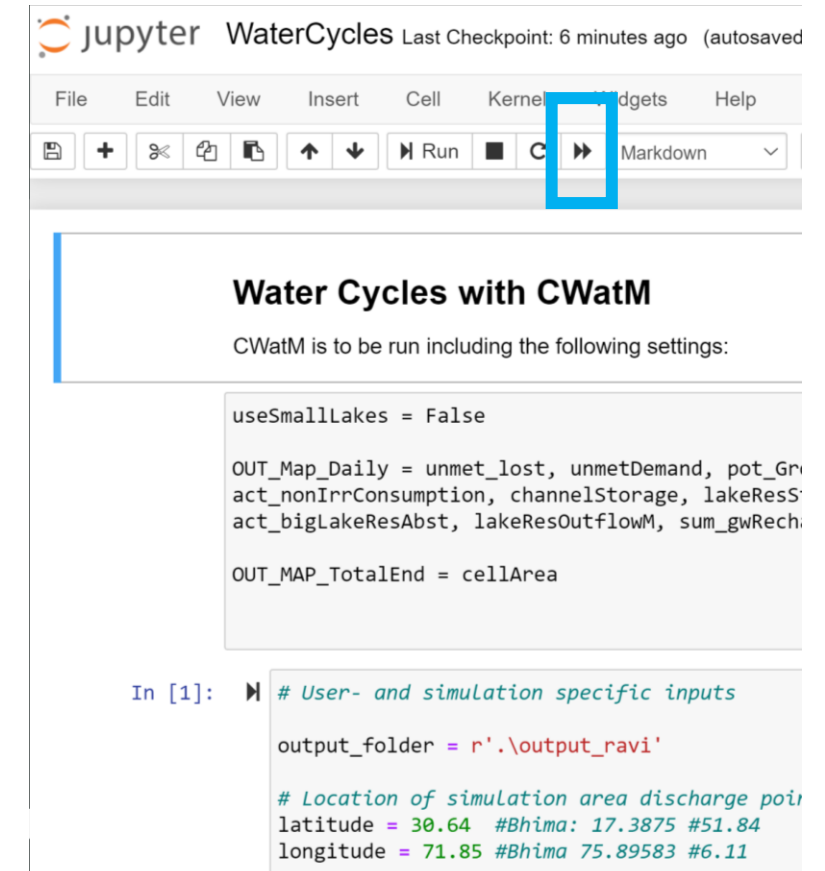
```
jupyter notebook
```

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved
C:\CWATM\CWATM_exercise6>jupyter notebook
```

2. Click WaterCycles.ipynb

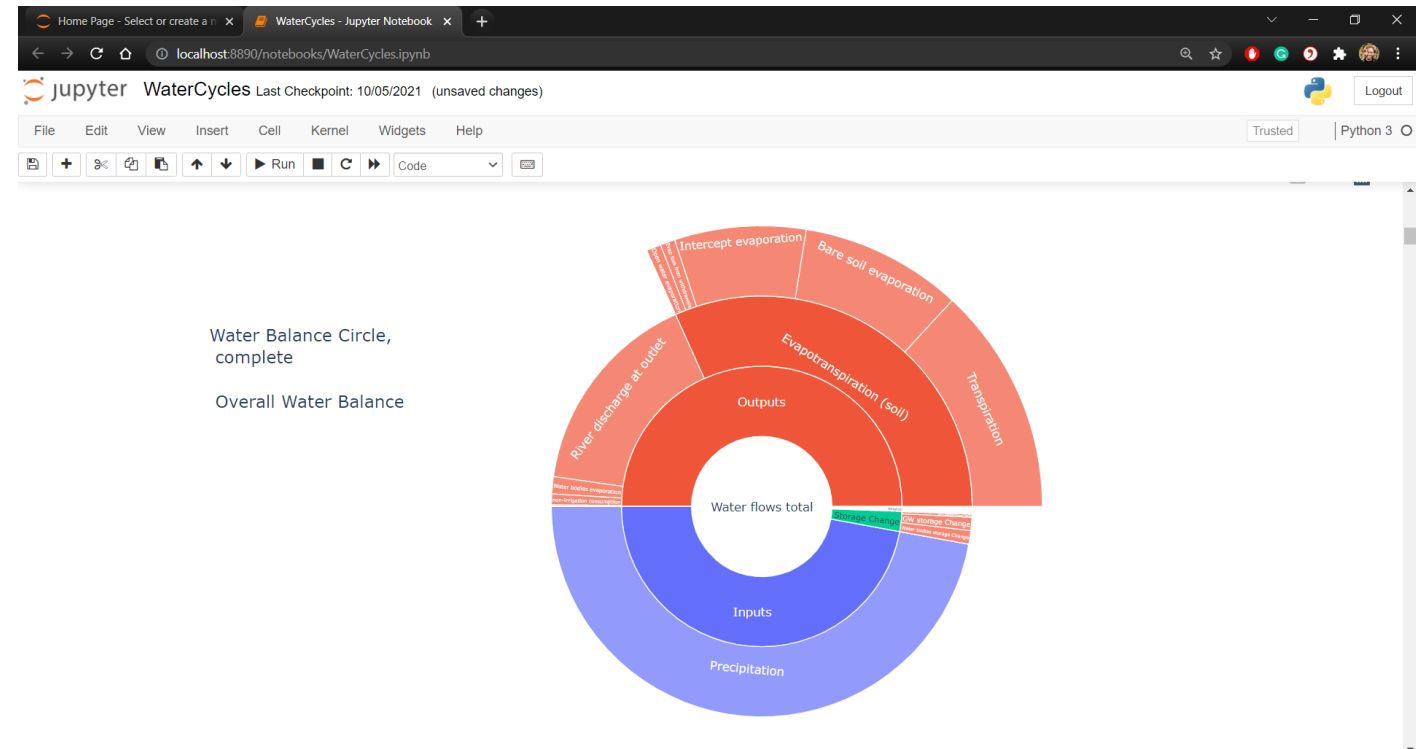


3. Execute the notebook



If the notebook has executed successfully, scroll down to see the collection of water cycles and signatures for an example basin.

Now to create these with your simulation data.



3. Play CWatM with these settings

Play CWatM with a familiar settings file, updated with the following settings:

```
MaskMap = Longitude Latitude (basin outlet)
Gauges = Longitude Latitude (basin outlet)
PathOut = ./output
useSmallLakes = False
limitAbstraction = False
OUT_Map_Daily = snowEvap, Rain, Snow, actTransTotal_forest,
actTransTotal_grasslands, actTransTotal_paddy,
actTransTotal_nonpaddy, unmet_lost, unmetDemand,
pot_GroundwaterAbstract, discharge, storGroundwater,
nonFossilGroundwaterAbs, Precipitation, totalET, EvapoChannel,
EvapWaterBodyM, act_nonIrrConsumption, channelStorage,
lakeResStorage, totalSto, sum_actTransTotal,
sum_actBareSoilEvap, sum_interceptEvap, sum_openWaterEvap,
addtoevapotrans, lakeResInflowM, act_bigLakeResAbst,
lakeResOutflowM, sum_gwRecharge, sum_capRiseFromGW, baseflow,
act_totalIrrConsumption, sum_runoff, returnFlow,
act_SurfaceWaterAbstract
OUT_MAP_TotalEnd = cellArea
```

Alternatively, the settings file *settings_WaterCycles.ini* already has the correct settings. Simply change MaskMap and Gauges to the coordinates of the outlet of any basin of interest.

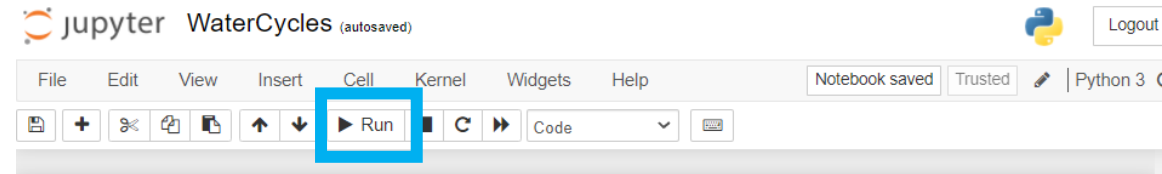
For example, for the Nile basin

MaskMap = 30.45 31.4

Gauges = 30.45 31.4

4. WaterCycles, new basin

In the Jupyter Notebook, update the **output folder** to the folder holding simulation results, and update the latitude and longitude **coordinates** to the associated basin outlet. Then, execute the Notebook.



Water Cycles with CWatM

Water cycles are a tool to evaluate simulations, illustrating balanced combinations of inflows, outflows, and storage changes as experienced through space and time in the simulation. This tool is recommended for simulations longer than one week and less than 4 years long.

CWatM is to be run including the following settings:

useSmallLakes = False

limitAbstraction = False

OUT_Map_Daily = snowEvap, Rain, Snow, actTransTotal_forest, actTransTotal_grasslands, actTransTotal_paddy, actTransTotal_nonpaddy, unmet_lost, unmetDemand, pot_GroundwaterAbstract, discharge, storGroundwater, nonFossilGroundwaterAbs, Precipitation, totalET, EvapoChannel, EvapWaterBodyM, act_nonIrrConsumption, channelStorage, lakeResStorage, totalSto, sum_actTransTotal, sum_actBareSoilEvap, sum_interceptEvap, sum_openWaterEvap, addtoevapotrans, lakeResInflowM, act_bigLakeResAbst, lakeResOutflowM, sum_gwRecharge, sum_capRiseFromGW, baseflow, act_totallrrConsumption, sum_runoff, returnFlow, act_SurfaceWaterAbstract

OUT_MAP_TotalEnd = cellArea

Update output folder and coordinates

```
In [1]: # User- and simulation specific inputs
```

```
output_folder = './output' |
```

```
# Location of simulation area discharge point
```

```
latitude = 17.3875
```

```
longitude = 75.89583
```

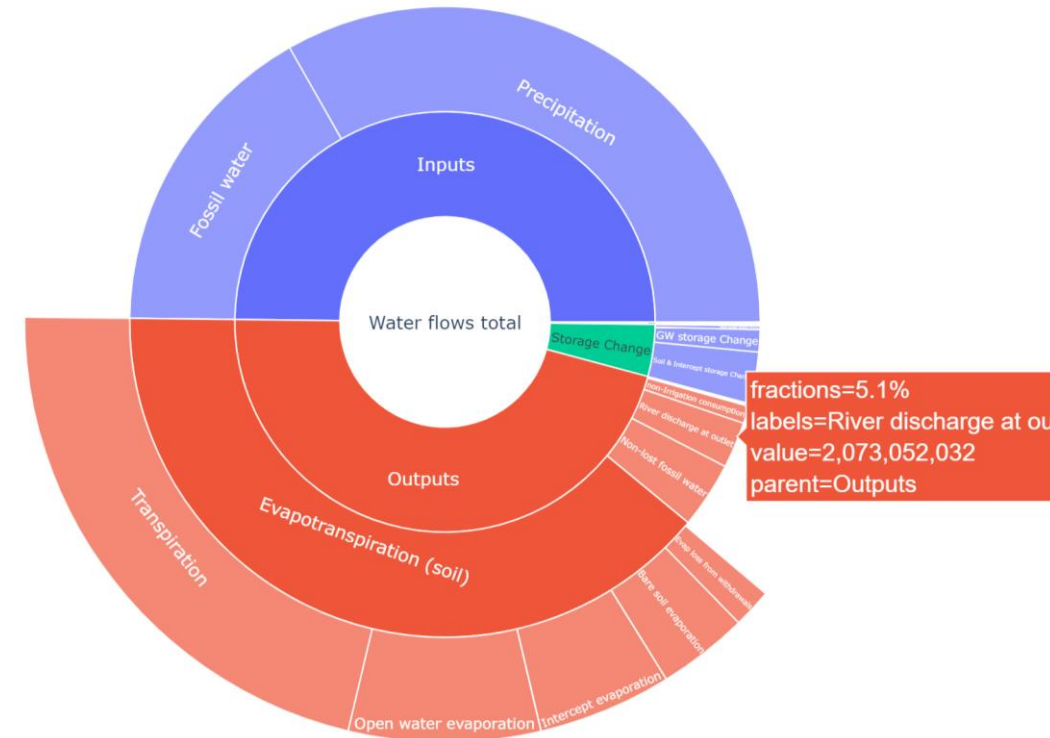
```
In [2]: run ./_functions_WaterCycles.ipynb
```

```
['storGroundwater', 'pot_GroundwaterAbstract', 'unmetDemand', 'Precipitation', 'totalE  
T', 'EvapoChannel', 'EvapWaterBodyM', 'act_nonIrrConsumption', 'channelStorage', 'lakeRe  
sStorage', 'totalSto', 'sum_actTransTotal', 'sum_actBareSoilEvap', 'sum_interceptEvap', 'sum_openWaterEvap', 'addtoevapotrans', 'lakeResInflowM', 'act_bigLakeResAbst', 'lakeResOutflowM', 'sum_gwRecharge', 'sum_capRiseFromGW', 'baseflow', 'act_totallrrConsumption', 'sum_runoff', 'returnFlow', 'act_SurfaceWaterAbstract']
```

Ravi subbasin

MaskMap = 71.85 30.64 (long lat)

Gauges = 71.85 30.64 (long lat)



StepStart = 01/04/2002

SpinUp = 01/05/2002

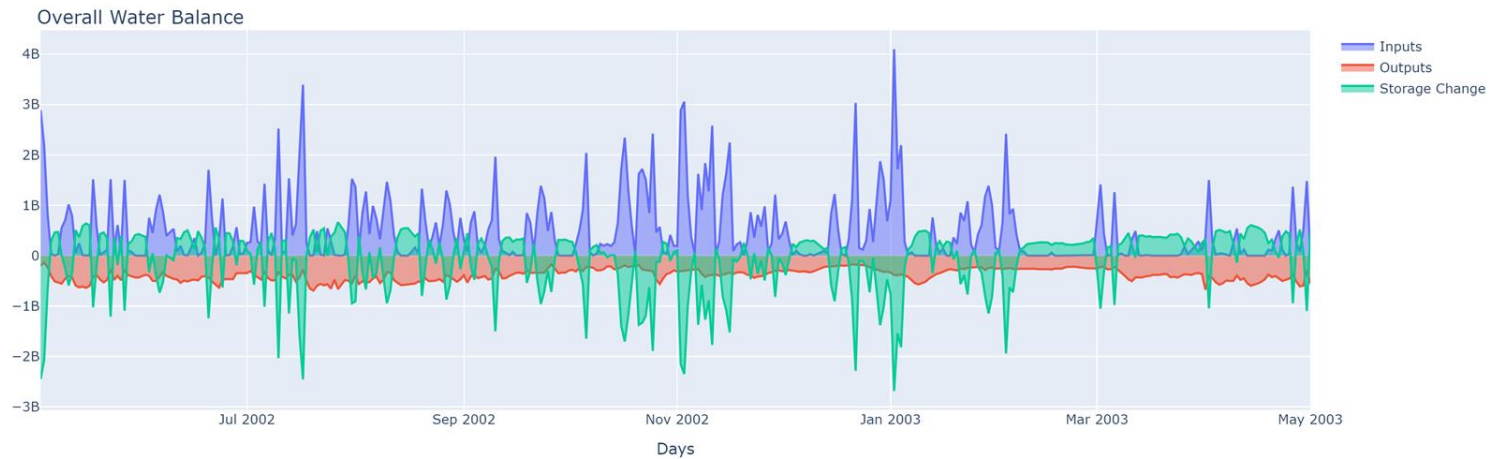
StepEnd = 01/05/2003

Rhine basin

MaskMap = 6.11 51.84

Gauges = 6.11 51.84

Water flows, daily



StepStart = 01/04/2002

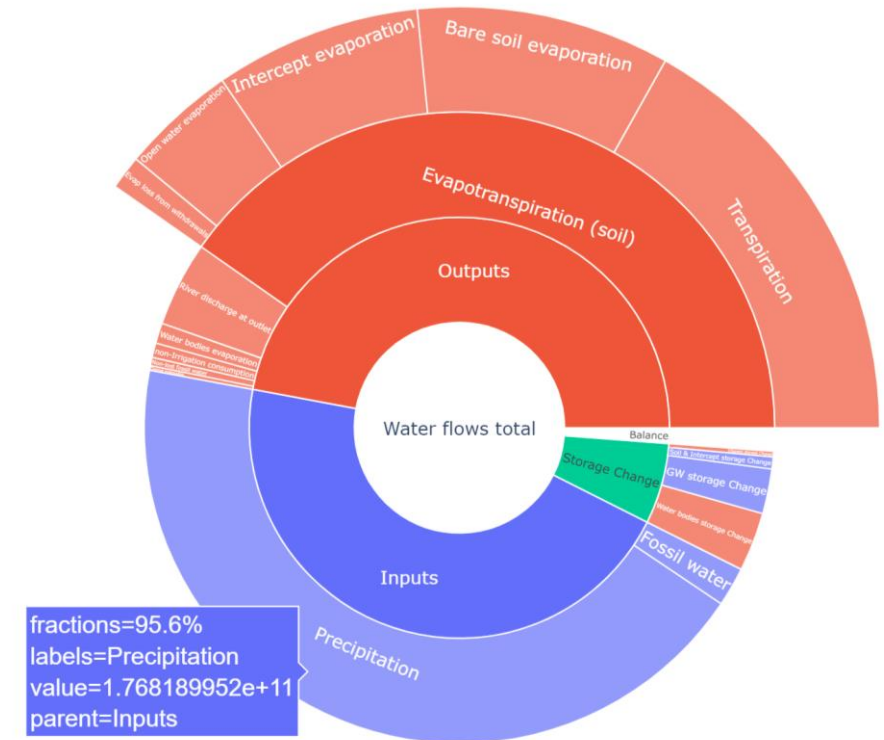
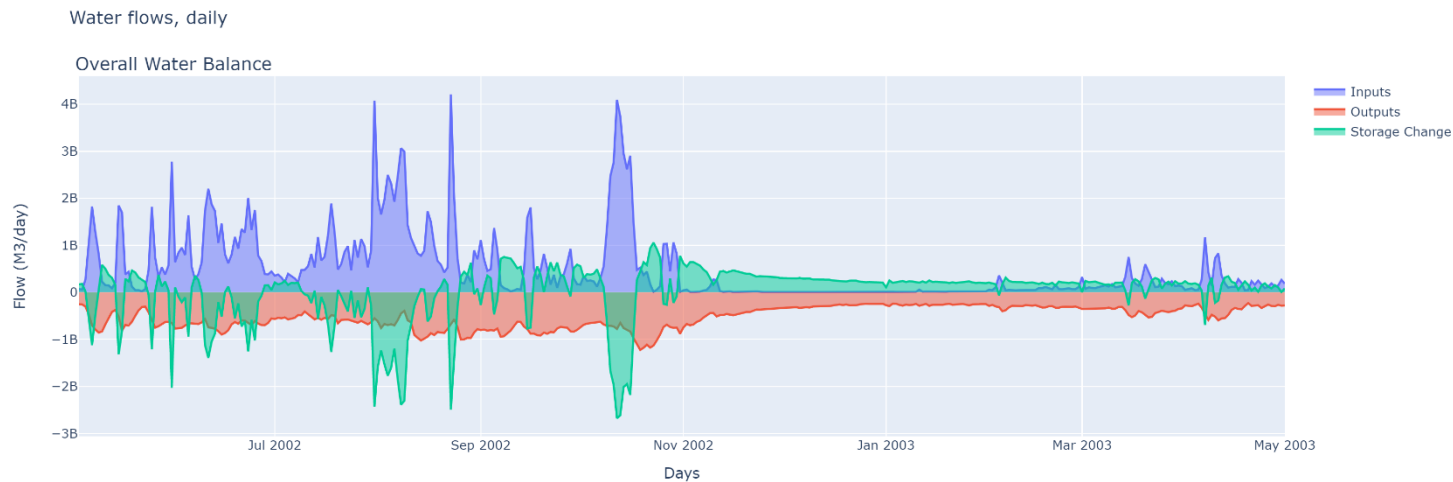
SpinUp = 01/05/2002

StepEnd = 01/05/2003

Krishna basin

MaskMap = 80.875 15.875

Gauges = 80.875 15.875



StepStart = 01/04/2002

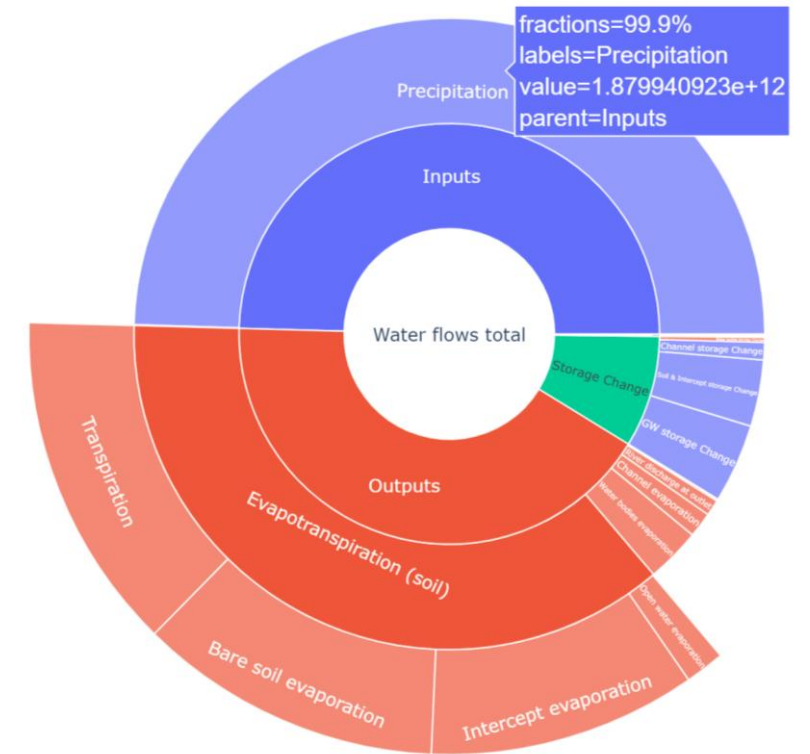
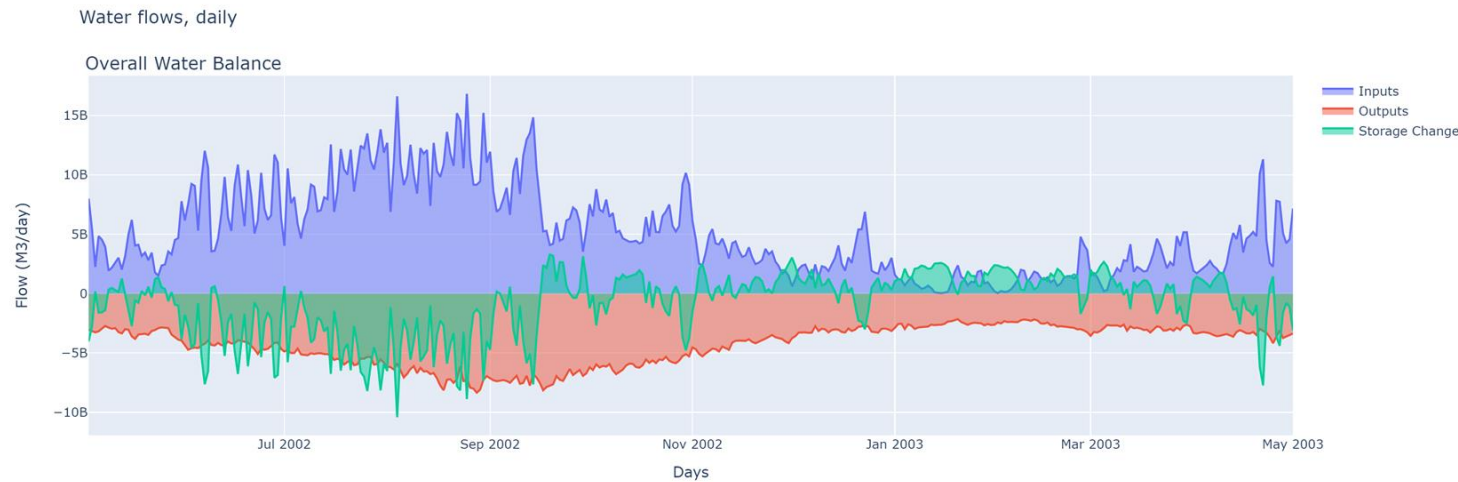
SpinUp = 01/05/2002

StepEnd = 01/05/2003

Nile basin

MaskMap = 30.45 31.4

Gauges = 30.45 31.4



StepStart = 01/04/2002

SpinUp = 01/05/2002

StepEnd = 01/05/2003