Australian Water School:
Python applications for Hydrology and Hydrogeology

Luk Peeters, Vincent Post, Chris Turnadge, Krey Price

Introduction

- Showcase of Python applications in hydrology and hydrogeology
  - Data wrangling
  - Visualisation
  - Data analysis
  - Pre –and post-processing models
- https://awschool.com.au/training/live-course-python-for-hydrology-hydrogeology/
- Jupyter Notebooks

Why Python / scripted language?

Automate repetitive tasks

Reproducible and repeatable

Self-documenting (Notebooks)

Heaps of packages

- Visualisation
- Data analysis & Machine learning

Free

- as in speech
- as in beer

CSIRO

Data wrangling

- bomwater
- https://pypi.org/project/bomwater/
- A python tool for requesting data from BoM Sensor Observation Service (SOS2, as WaterML 2.0 format)

jupyter Access_BoM_Water_Service (unsaved changes)

Visit repo | Copy Binder link

File Edit View Insert Cell Kernel Widgets Help | Trusted | Python 3

Run | Code | Voilà | Download | GitHub | Binder | Memory: 541.1 MB / 2 GB

In [1]:
```
import bom_water
from sidecar import Sidecar
from ipyleaflet import Map, GeoJSON, Polygon
from IPython.display import display, clear_output, HTML, JSON
import json
import requests
```

Get Capability

In [2]:
```
bm = bom_water.BomWater()
response = bm.request(bm.actions.GetCapabilities)
# response
JSON(bm.xml_to_json( response.text))
```

```
one time creating cache directory and files, this will take a little time please wait.
finished creating cache directory and files
```

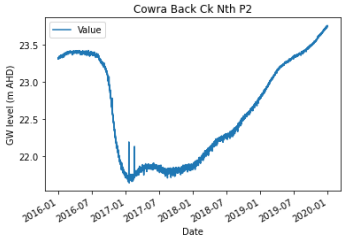
Out[2]:
```
<IPython.core.display.JSON object>
```

Get Observation

Return data type is a Pandas DataFrame

In [7]:
```
t_begin = "2016-01-01T00:00:00+10"
t_end = "2019-12-31T00:00:00+10"

try:
    response = bm.request(bm.actions.GetObservation, bm.features.Cowra_Back_Ck_Nth_P2, bm.properties.Ground_Water_Level, bm.proce
except response.exceptions.RequestException as e:
    display(f'BoM service failed with RequestException: {e}')

ts = bm.parse_get_data(response)
ts.plot.line()
import matplotlib.pyplot as plt
plt.title('Cowra Back Ck Nth P2')
plt.xlabel('Date')
plt.ylabel('GW level (m AHD)')
```

Out[7]:
```
Text(0, 0.5, 'GW level (m AHD)')
```



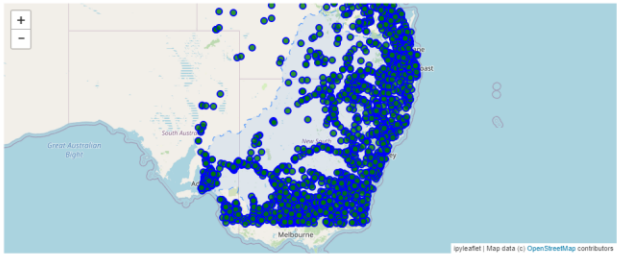
Data wrangling

- bomwater
- https://pypi.org/project/bomwater/
- A python tool for requesting data from BoM Sensor Observation Service (SOS2, as WaterML 2.0 format)

Get Feature of interest

Simple demo of requesting features within a bounding box and mapping their locations

In [5]:
```
low_left_lat = -37.505032
low_left_long = 138.00
upper_right_lat = -24.00
upper_right_long = 154.00

lower_left_coords = f'{low_left_lat} {low_left_long}'
upper_right_coords = f'{upper_right_lat} {upper_right_long}'

bm = bom_water.BomWater()

# request( action, feature=None, prop=None, proced=None, begin=None, end=None, lower_corner=None, upper_corner=None)

response = bm.request(bm.actions.GetFeatureOfInterest, None, bm.properties.Water_Course_Discharge, bm.procedures.Pat4_C_B_1_Daily
response_json = bm.xml_to_json(response.text)
'''bomwater creates a FeatureCollection which can be used for mapping'''
feature_list = bm.create_feature_list(response_json, None )

bbox_poly = Polygon(
        locations=[(low_left_lat, low_left_long),(low_left_lat, upper_right_long),(upper_right_lat, upper_right_long),(upper_rig
    color="green",
    fill_color="green"
        )
m = Map(center=(-32, 146), zoom=5)

geojsonMarkerOptions = {
    'radius': 8,
    'fillColor': "#ff7800",
    'color': "#000",
    'weight': 1,
    'opacity': 1,
    'fillOpacity': 0.8
};

geo_json_stations = GeoJSON(
    data=feature_list,
    point_style={'radius': 5, 'color': 'blue', 'fillOpacity': 0.8, 'fillColor': 'green', 'weight': 2})

loci_mdb = "https://gds.loci.cat/geometry/geofabric2_1_1_awradrainagedivision/94002062?_format=application/json&_view=simplifiedge
def random_color(feature):
    return {
        'color': 'black',
        'fillColor': 'green',#random.choice(['red', 'yellow', 'green', 'orange']),
    }
mdb = json.loads(requests.get(loci_mdb).text)
geo_json_mdb = GeoJSON(
    data=mdb,
    style={
        'opacity': 1, 'dashArray': '9', 'fillOpacity': 0.1, 'weight': 1
    },
    hover_style={
        'color': 'green', 'dashArray': '0', 'fillOpacity': 0.3
    },
    style_callback=random_color
)

m.add_layer(geo_json_mdb)
# m.add_layer(bbox_poly)
m.add_layer(geo_json_stations)

sc = Sidecar(title='MDB sites')
with sc:
    display(m)

'''Display the json response'''
JSON(response_json)
```



Out[5]: <IPython.core.display.JSON object>

NumPy

pandas

matplotlib

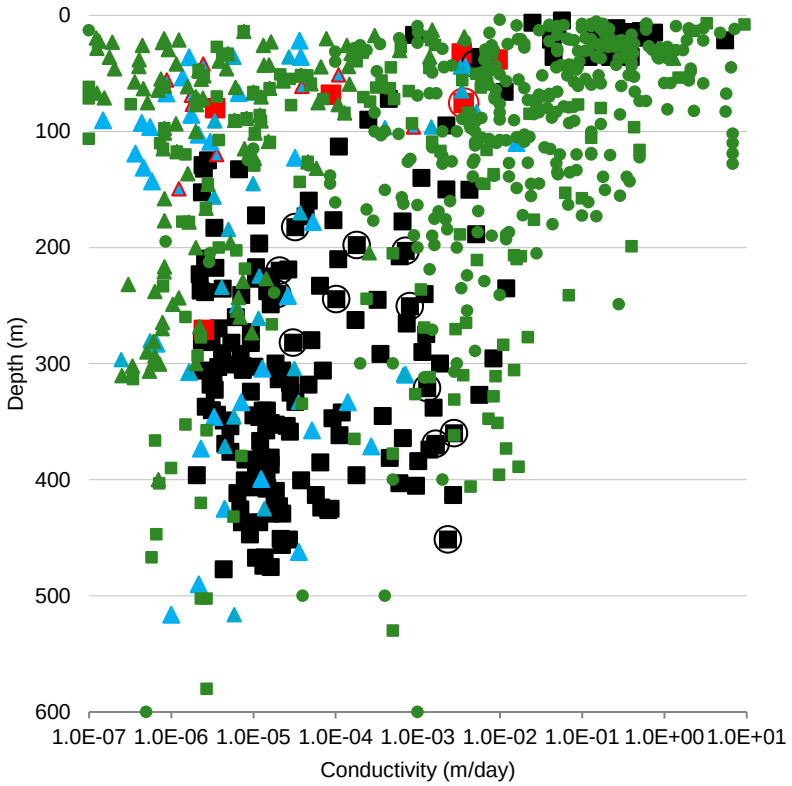
Version 3.4.1

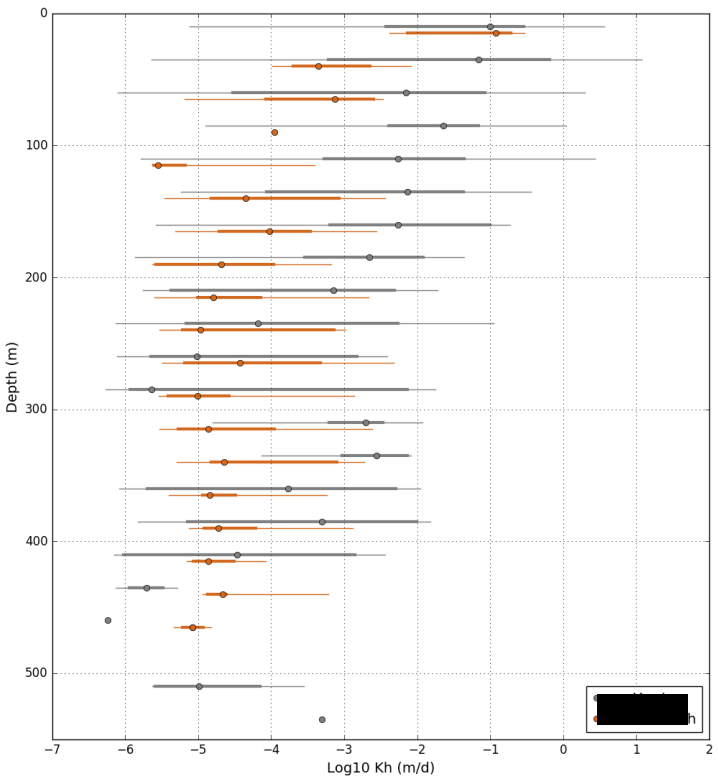
GeoPandas

scikit learn

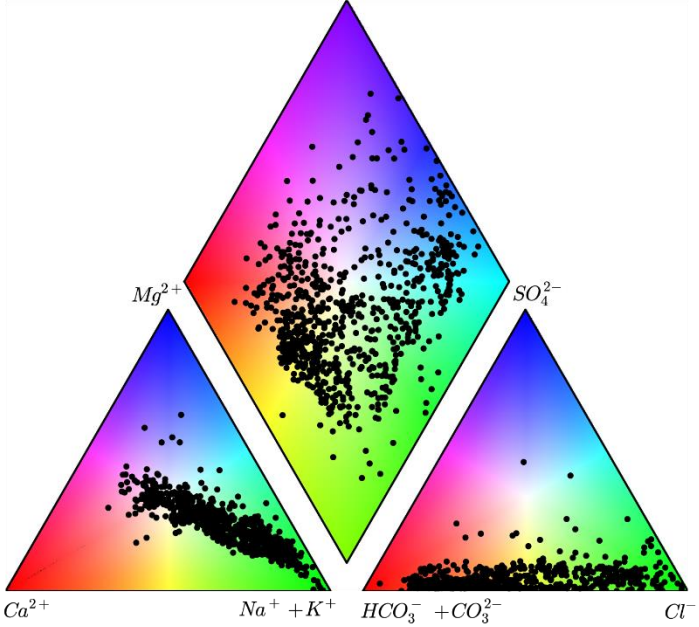
CSIRO

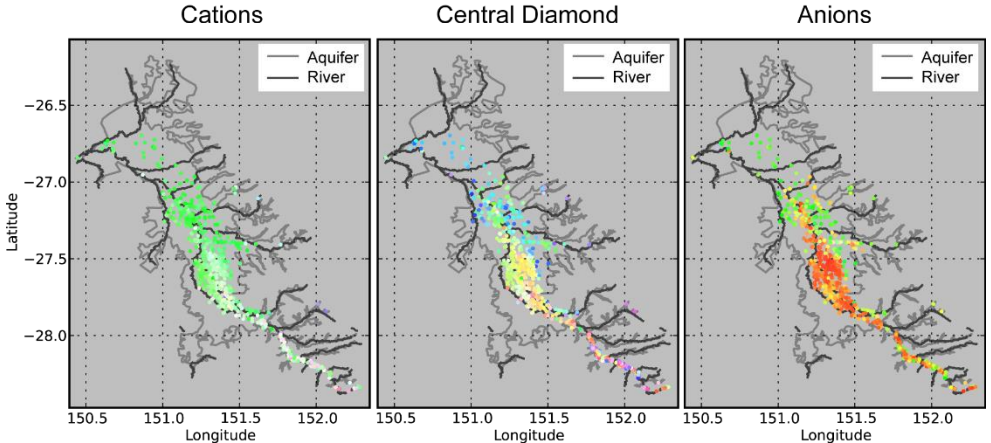
Let your data speak





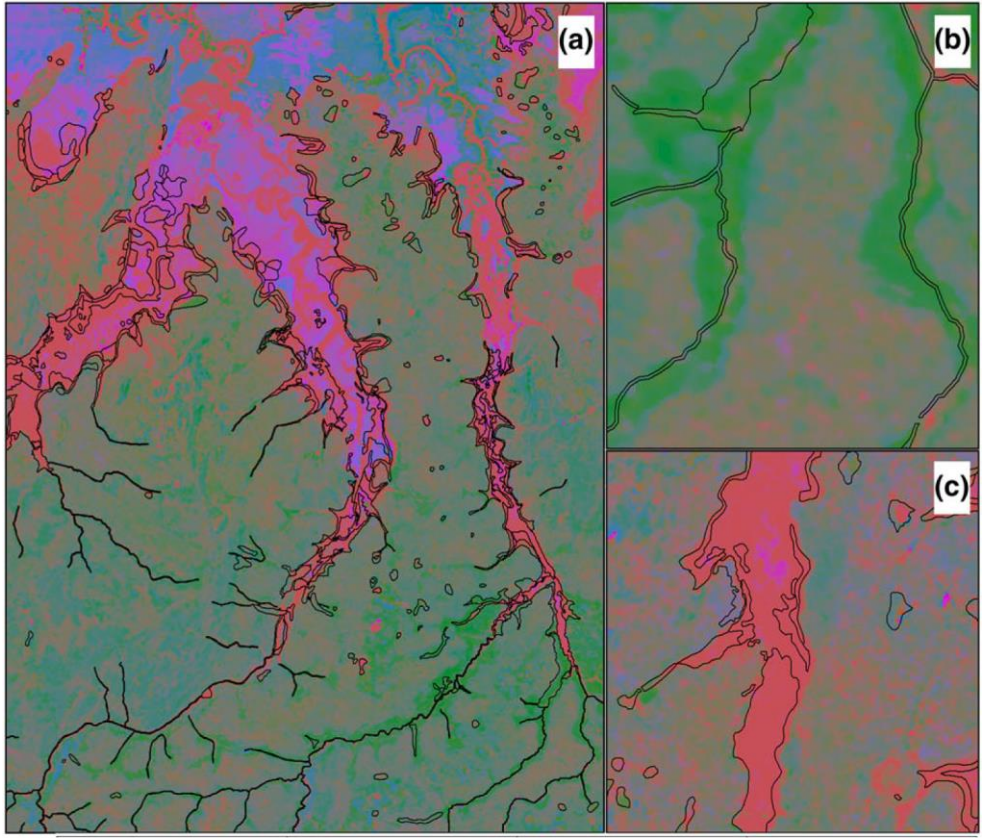
Using colour

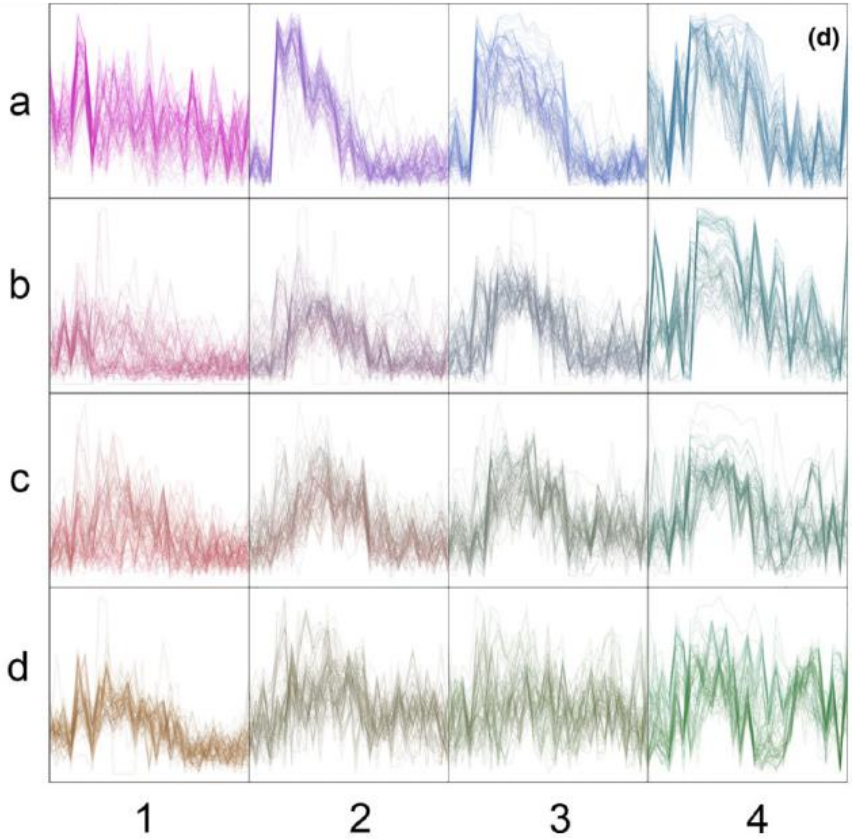




Peeters L (2013) A Background Color Scheme for Piper Plots to Spatially Visualize Hydrochemical Patterns. Groundwater 52(1), 2--6. Doi: 10.1111/gwat.12118.

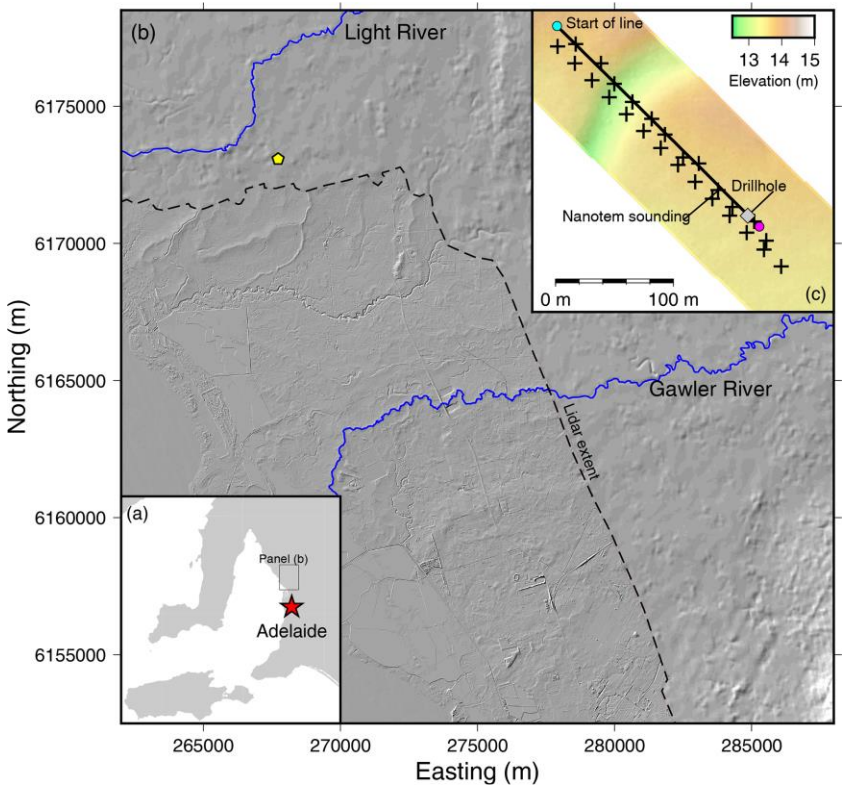
Using colour and machine learning

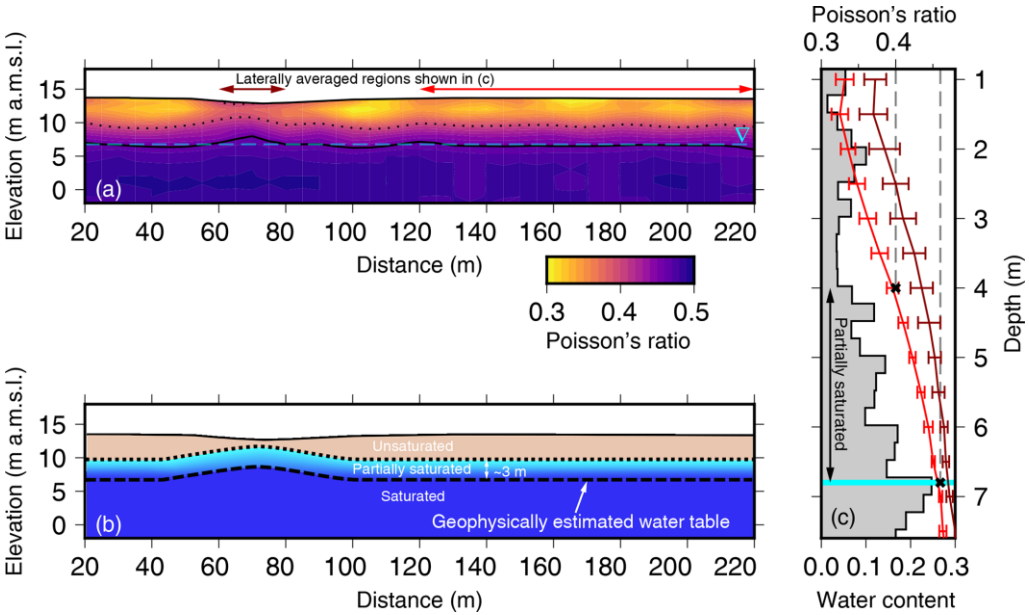




Castellazzi P, Doody T and Peeters L (2019) Toward monitoring groundwater-dependant ecosystems using SAR imagery. Hydrological Processes, hy 13570. Doi: 10.1002/hyp.13570.

Geophysical inversion & processing: PyGIMli

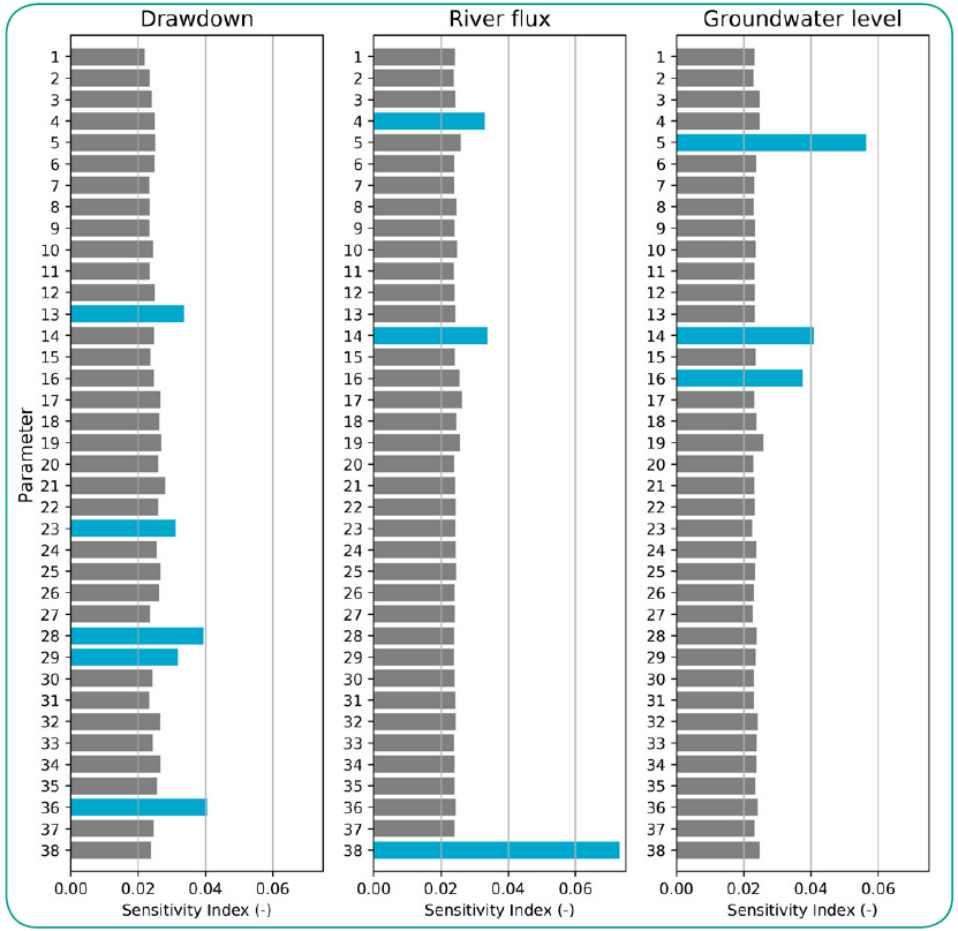




Flinchum B, Banks E, Hatch M, Batelaan O, Peeters L and Pasquet S (2020) Identifying recharge under subtle ephemeral features in flat-lying semi-arid region using a combined geophysical approach. Hydrology And Earth System Sciences 24, 4353--4368. Doi: 10.5194/hess-24-4353-2020.

Processing model results: sensitivity analysis

- SALib
- Sensitivity analysis library in Python
- https://salib.readthedocs.io/en/latest/



Peeters LJM, Crosbie RS, Henderson BL, Holland K, Lewis S, Post DA and Schmidt RK (2018) The importance of being uncertain. Water e-Journal 3(2), 10. Doi: 10.21139/wej.2018.018.