06_Step6_Sensitivity of Bear and Weber WEAP Models System's Reliability due to changes in input

By Adel M. Abdallah, Feb 2022

Execute the following cells by pressing `Shift-Enter`, or by pressing the play button on the toolbar above.

1. Import python libraries

In [17]:
```
# 1. Import python libraries
### set the notebook mode to embed the figures within the cell
import numpy
import sqlite3
import numpy as np
import pandas as pd
import getpass
from hs_restclient import HydroShare, HydroShareAuthBasic
import os

import plotly
plotly.__version__
import plotly.offline as offline
import plotly.graph_objs as go
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
offline.init_notebook_mode(connected=True)
from plotly.offline import init_notebook_mode, iplot
from plotly.graph_objs import *

init_notebook_mode(connected=True)         # initiate notebook for offline plot

import os
import csv
from collections import OrderedDict
import sqlite3
import pandas as pd
import numpy as np
from IPython.display import display, Image, SVG, Math, YouTubeVideo
import urllib
import calendar

print 'The needed Python libraries have been imported'
```

```
The needed Python libraries have been imported
```

1. Create a copy of the original Bear River WEAP Area to use while keeping the orignial as-as for any later use

You can always use this orignal one and delete any new copies you make afterwards.

In [18]:
```
# this library is needed to connect to the WEAP API
import win32com.client

# this command will open the WEAP software (if closed) and get the last active model
# you could change the active area to another one inside WEAP or by passing it to the command here
#WEAP.ActiveArea = "BearRiverFeb2017_V10.9"


WEAP=win32com.client.Dispatch("WEAP.WEAPApplication")

# WEAP.Visible = 'FALSE'

#-------------------------------------------------------------------

print WEAP.ActiveArea.Name
WEAP.ActiveArea = "Bear_River_WEAP_Model_2017_Original"
print WEAP.ActiveArea.Name

WEAP.Areas("Bear_River_WEAP_Model_2017_Original").Open
WEAP.ActiveArea = "Bear_River_WEAP_Model_2017_Original"
print WEAP.ActiveArea.Name
#-------------------------------------------------------------------


print 'Connected to WEAP API and the '+ WEAP.ActiveArea.Name + ' Area'
print '-------------'
if not WEAP.Registered:
    print "Because WEAP is not registered, you cannot use the API"

# get the active WEAP Area (model) to serve data into it

# ActiveArea=WEAP.ActiveArea.Name


# get the active WEAP scenario to serve data into it
print '-------------'

ActiveScenario= WEAP.ActiveScenario.Name
print '\n ActiveScenario= '+ActiveScenario
print '-------------'

WEAP_Area_dir=WEAP.AreasDirectory
print WEAP_Area_dir


print "\n \n You're connected to the WEAP API"


# Delete the Area if it exists and then add it. Start from fresh
Area="Bear_River_WEAP_Model_2017_sedimentation"

if not WEAP.Areas.Exists(Area):
    WEAP.SaveAreaAs(Area)
WEAP.Quit
print " Created the Bear_River_WEAP_Model_2017_sedimentation Area"

WEAP=win32com.client.Dispatch("WEAP.WEAPApplication")

#-------------------------------------------------------------------
print WEAP.ActiveArea.Name
WEAP.ActiveArea = "Bear_River_WEAP_Model_2017_Original"
print WEAP.ActiveArea.Name

WEAP.Areas("Bear_River_WEAP_Model_2017_Original").Open
WEAP.ActiveArea = "Bear_River_WEAP_Model_2017_Original"
print WEAP.ActiveArea.Name
#-------------------------------------------------------------------


# Delete the Area if it exists and then add it. Start from fresh
Area="Bear_River_WEAP_Model_2017_scenarios"

if not WEAP.Areas.Exists(Area):
    WEAP.SaveAreaAs(Area)
print  'ActiveArea= '+ WEAP.ActiveArea.Name


WEAP.Areas("Bear_River_WEAP_Model_2017_scenarios").Open
WEAP.ActiveArea = "Bear_River_WEAP_Model_2017_scenarios"
print WEAP.ActiveArea.Name

Scenarios=[]
Scenarios=['Evaporation','Headflow','Demand']

for Scenario in Scenarios:
    if WEAP.Scenarios.Exists(Scenario):
        # delete it
        WEAP.Scenarios(Scenario).Delete(True)
        # add it back as a fresh copy
        WEAP.Scenarios.Add(Scenario,'Reference')
    else:
        WEAP.Scenarios.Add(Scenario,'Reference')

WEAP.ActiveArea.Save
WEAP.SaveArea
# Make a copy from the reference (base) scenario
# WEAP.Scenarios.Add('UpdateCacheDemand','Reference')
print '---------------------- \n'
print 'Scenarios added to the Bear_River_WEAP_Model_2017_scenarios WEAP area'
```

```
Bear_River_WEAP_Model_2017_scenarios
Bear_River_WEAP_Model_2017_scenarios
Bear_River_WEAP_Model_2017_Original
Connected to WEAP API and the Bear_River_WEAP_Model_2017_Original Area
-------------
-------------

 ActiveScenario= Current Accounts
-------------
C:\Users\Rosenberg\Documents\WEAP Areas\


 You're connected to the WEAP API
 Created the Bear_River_WEAP_Model_2017_sedimentation Area
Bear_River_WEAP_Model_2017_Original
Bear_River_WEAP_Model_2017_Original
Bear_River_WEAP_Model_2017_Original
ActiveArea= Bear_River_WEAP_Model_2017_Original
Bear_River_WEAP_Model_2017_scenarios
----------------------

Scenarios added to the Bear_River_WEAP_Model_2017_scenarios WEAP area
```

2. Create a copy of the original Weber River WEAP Area to use while keeping the orignial as-as for any later use

You can always use this orignal one and delete any new copies you make afterwards.Get the Weber scenarios

In [21]:
```
# this library is needed to connect to the WEAP API
import win32com.client

# this command will open the WEAP software (if closed) and get the last active model
# you could change the active area to another one inside WEAP or by passing it to the command here
#WEAP.ActiveArea = "BearRiverFeb2017_V10.9"


WEAP=win32com.client.Dispatch("WEAP.WEAPApplication")

# WEAP.Visible = 'FALSE'

#-------------------------------------------------------------------

print WEAP.ActiveArea.Name
WEAP.ActiveArea = "WeberOgdenRiversLab-3_Original"
print WEAP.ActiveArea.Name

WEAP.Areas("WeberOgdenRiversLab-3_Original").Open
WEAP.ActiveArea = "WeberOgdenRiversLab-3_Original"
print WEAP.ActiveArea.Name
#-------------------------------------------------------------------


print 'Connected to WEAP API and the '+ WEAP.ActiveArea.Name + ' Area'
print '-------------'
if not WEAP.Registered:
    print "Because WEAP is not registered, you cannot use the API"

# get the active WEAP Area (model) to serve data into it

# ActiveArea=WEAP.ActiveArea.Name


# get the active WEAP scenario to serve data into it
print '-------------'

ActiveScenario= WEAP.ActiveScenario.Name
print '\n ActiveScenario= '+ActiveScenario
print '-------------'

WEAP_Area_dir=WEAP.AreasDirectory
print WEAP_Area_dir


print "\n \n You're connected to the WEAP API"


# Delete the Area if it exists and then add it. Start from fresh
Area="WeberOgdenRiversLab-3_sedimentation"

if not WEAP.Areas.Exists(Area):
    WEAP.SaveAreaAs(Area)
WEAP.Quit
print " Created the WeberOgdenRiversLab-3_sedimentation Area"

WEAP=win32com.client.Dispatch("WEAP.WEAPApplication")

#-------------------------------------------------------------------
print WEAP.ActiveArea.Name
WEAP.ActiveArea = "WeberOgdenRiversLab-3_Original"
print WEAP.ActiveArea.Name

WEAP.Areas("WeberOgdenRiversLab-3_Original").Open
WEAP.ActiveArea = "WeberOgdenRiversLab-3_Original"
print WEAP.ActiveArea.Name
#-------------------------------------------------------------------


# Delete the Area if it exists and then add it. Start from fresh
Area="WeberOgdenRiversLab-3_scenarios"

if not WEAP.Areas.Exists(Area):
    WEAP.SaveAreaAs(Area)
print  'ActiveArea= '+ WEAP.ActiveArea.Name


WEAP.Areas("WeberOgdenRiversLab-3_scenarios").Open
WEAP.ActiveArea = "WeberOgdenRiversLab-3_scenarios"
print WEAP.ActiveArea.Name

Scenarios=[]
Scenarios=['Evaporation','Headflow','Demand']

for Scenario in Scenarios:
    if WEAP.Scenarios.Exists(Scenario):
        # delete it
        WEAP.Scenarios(Scenario).Delete(True)
        # add it back as a fresh copy
        WEAP.Scenarios.Add(Scenario,'Reference')
    else:
        WEAP.Scenarios.Add(Scenario,'Reference')

WEAP.ActiveArea.Save
WEAP.SaveArea
# Make a copy from the reference (base) scenario
# WEAP.Scenarios.Add('UpdateCacheDemand','Reference')
print '---------------------- \n'
print 'Scenarios added to the WeberOgdenRiversLab-3_scenarios WEAP area'
```

```
Bear_River_WEAP_Model_2017_sedimentation
WeberOgdenRiversLab-3_Original
WeberOgdenRiversLab-3_Original
Connected to WEAP API and the WeberOgdenRiversLab-3_Original Area
-------------
-------------

 ActiveScenario= Reference
-------------
C:\Users\Rosenberg\Documents\WEAP Areas\


 You're connected to the WEAP API
 Created the WeberOgdenRiversLab-3_sedimentation Area
WeberOgdenRiversLab-3_Original
WeberOgdenRiversLab-3_Original
WeberOgdenRiversLab-3_Original
ActiveArea= WeberOgdenRiversLab-3_Original
WeberOgdenRiversLab-3_scenarios
----------------------

Scenarios added to the WeberOgdenRiversLab-3_scenarios WEAP area
```

Save WEAP and close the connection

In [ ]:
```
# 9. Close the SQLite and WEAP API connections
conn.close()

print 'connection disconnected'

# Uncomment
WEAP.SaveArea



# this command will close WEAP
WEAP.Quit

print 'Connection with WEAP API is disconnected'
```

Prepare the scenarios input data in WaMDaM

Shortcut I already prepared the scenarios data for you in WaMDaM Workbooks and loaded them to SQLite databse.

You can view the WaMDaM workbooks here https://github.com/WamdamProject/WaMDaM_JupyteNotebooks\2_ServeToModels\Files

Click the **Connect to SQLite** tab (Figure 1), then click the button **Connect to an Existing SQLite WaMDaM database**

From the previous step, it is expected that you already have clones the GitHub repo https://github.com/WamdamProject/WaMDaM_JupyterNotebooks

Navigate to the location on your desktop where you have the GitHub clones folder. For example:

```
C:\Users\Adel\Documents\GitHub\WamdamProject\WaMDaM_JupyteNotebooks\2_ServeToModels\
```

Connect to the SQLite file Bear_Weber.sqlite

Congratulations!