# 05_Step5_Summary and comparisons of Bear and Weber WEAP Models

**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 [1]:
```
# 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
```

## 2. Connect to the WaMDaM populated SQLite file

In [2]:
```
# 2. Connect to the WaMDaM populated SQLite file


# Then we can run queries against it within this notebook :)

# the SQLite file is published here
#https://github.com/WamdamProject/WaMDaM_UseCases/blob/master/UseCases_files/3SQLite_database/BearRiverDataset
WaMDaM_SQLite_Name='Bear_Weber.sqlite'

# WaMDaM_SQLite_Name='replicateWaMDaM.sqlite'


conn = sqlite3.connect(WaMDaM_SQLite_Name)

print 'Connected to the WaMDaM SQLite file called'+': '+ WaMDaM_SQLite_Name
```

```
Connected to the WaMDaM SQLite file called: Bear_Weber.sqlite
```

## Summary and comparisons of Bear and Weber WEAP Models

In [3]:
```
# Use Case 3.1Identify_aggregate_TimeSeriesValues.csv
# plot aggregated to monthly and converted to acre-feet time series data of multiple sources



# 2.2Identify_aggregate_TimeSeriesValues.csv
Query_UseCase3_1_URL="""
https://raw.githubusercontent.com/WamdamProject/WaMDaM_JupyterNotebooks/master/2_ServeToModels/SQL/GetModelSum
"""

# Read the query text inside the URL
Query_UseCase3_1_text = urllib.urlopen(Query_UseCase3_1_URL).read()


# return query result in a pandas data frame
result_df_UseCase3_1= pd.read_sql_query(Query_UseCase3_1_text, conn)

# uncomment the below line to see the list of attributes
display (result_df_UseCase3_1)


print "Queries are done"
```

| | MasterNetworkName | ScenarioName | AttributeDataTypeCV | CountOfAttributes | CountOfInstances |
|---|---|---|---|---|---|
| **0** | Bear River Network | Bear River WEAP Model 2017 | FreeText | 89 | 61 |
| **1** | Bear River Network | Bear River WEAP Model 2017 | MultiAttributeSeries | 11 | 11 |
| **2** | Bear River Network | Bear River WEAP Model 2017 | NumericValues | 281 | 197 |
| **3** | Bear River Network | Bear River WEAP Model 2017 | SeasonalNumericValues | 36 | 33 |
| **4** | Bear River Network | Bear River WEAP Model 2017 | TimeSeries | 37 | 36 |
| **5** | Weber | Base | FreeText | 32 | 30 |
| **6** | Weber | Base | MultiAttributeSeries | 8 | 8 |
| **7** | Weber | Base | NumericValues | 464 | 152 |
| **8** | Weber | Base | SeasonalNumericValues | 15 | 15 |
| **9** | Weber | Base | TimeSeries | 44 | 44 |

```
Queries are done
```

## Total demand

In [4]:
```
# Use Case 3.1Identify_aggregate_TimeSeriesValues.csv
# plot aggregated to monthly and converted to acre-feet time series data of multiple sources



# 2.2Identify_aggregate_TimeSeriesValues.csv
Query_UseCase3_1_URL="""
https://raw.githubusercontent.com/WamdamProject/WaMDaM_JupyterNotebooks/master/2_ServeToModels/SQL/GetAverageA
"""

# Read the query text inside the URL
Query_UseCase3_1_text = urllib.urlopen(Query_UseCase3_1_URL).read()


# return query result in a pandas data frame
result_df_UseCase3_1= pd.read_sql_query(Query_UseCase3_1_text, conn)

# uncomment the below line to see the list of attributes
display (result_df_UseCase3_1)



print "Queries are done"
```

| | ScenarioName | AttributeDataTypeCV | NoumInstances | AverageAnnualDemand |
|---|---|---|---|---|
| **0** | Base | TimeSeries | 19 | 473385.212121 |
| **1** | Bear River WEAP Model 2017 | NumericValues | 3 | 77799.090000 |
| **2** | Bear River WEAP Model 2017 | SeasonalNumericValues | 21 | 990552.939225 |

```
Queries are done
```

## Total Discharge

In [5]:
```
# Use Case 3.1Identify_aggregate_TimeSeriesValues.csv
# plot aggregated to monthly and converted to acre-feet time series data of multiple sources


# 2.2Identify_aggregate_TimeSeriesValues.csv
Query_UseCase3_1_URL="""
https://raw.githubusercontent.com/WamdamProject/WaMDaM_JupyterNotebooks/master/2_ServeToModels/SQL/GetTotalHe
"""

# Read the query text inside the URL
Query_UseCase3_1_text = urllib.urlopen(Query_UseCase3_1_URL).read()

# return query result in a pandas data frame
result_df_UseCase3_1= pd.read_sql_query(Query_UseCase3_1_text, conn)

# uncomment the below line to see the list of attributes
display (result_df_UseCase3_1)


print "Queries are done"
```

| | ResourceTypeAcronym | ScenarioName | #Instance | AttributeDataTypeCV | AttributeName_Abstract | AnnualDischarge |
|---|---|---|---|---|---|---|
| **0** | WEAP_Bear | Bear River WEAP Model 2017 | 11 | TimeSeries | Surface Water Inflow | 1.265191e+06 |
| **1** | WEAP_Bear | Bear River WEAP Model 2017 | 12 | TimeSeries | Headflow | 1.026013e+06 |
| **2** | WEAP_Bear | Bear River WEAP Model 2017 | Dewitt Spring Headflow | SeasonalNumericValues | Headflow | 1.059976e+04 |
| **3** | WEAP_Weber | Base | 17 | TimeSeries | Headflow | 9.423588e+05 |
| **4** | WEAP_Weber | Base | Q68 Sewage Outfall Headflow | SeasonalNumericValues | Headflow | 3.314313e+04 |

```
Queries are done
```

1 Cubic Feet Per Second to Acre-feet Per Month = 60.3707

## Total Capacity

In [6]:
```
# Use Case 3.1Identify_aggregate_TimeSeriesValues.csv
# plot aggregated to monthly and converted to acre-feet time series data of multiple sources

# 2.2Identify_aggregate_TimeSeriesValues.csv
Query_UseCase3_1_URL="""
https://raw.githubusercontent.com/WamdamProject/WaMDaM_JupyterNotebooks/master/2_ServeToModels/SQL/GetTotalRes
"""

# Read the query text inside the URL
Query_UseCase3_1_text = urllib.urlopen(Query_UseCase3_1_URL).read()


# return query result in a pandas data frame
result_df_UseCase3_1= pd.read_sql_query(Query_UseCase3_1_text, conn)

# uncomment the below line to see the list of attributes
display (result_df_UseCase3_1)


print "Queries are done"
```

| | ScenarioName | #Reservoirs | TotalCapacity |
|---|---|---|---|
| **0** | Base | 8 | 551240.000 |
| **1** | Bear River WEAP Model 2017 | 10 | 140411.002 |
| **2** | Bear River WEAP Model 2017 | Bear Lake | 1516633.000 |

```
Queries are done
```

In [ ]:
```

```