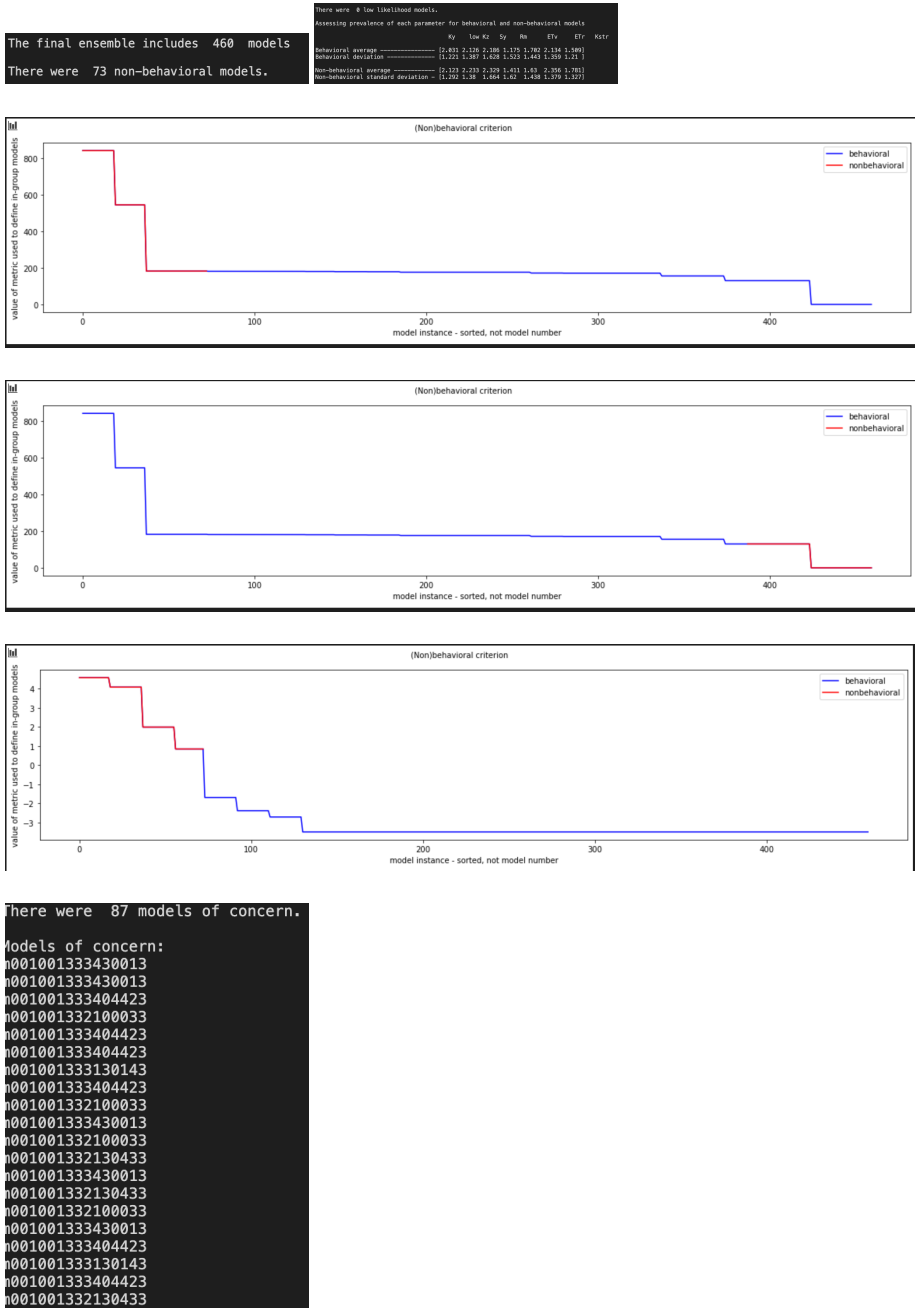


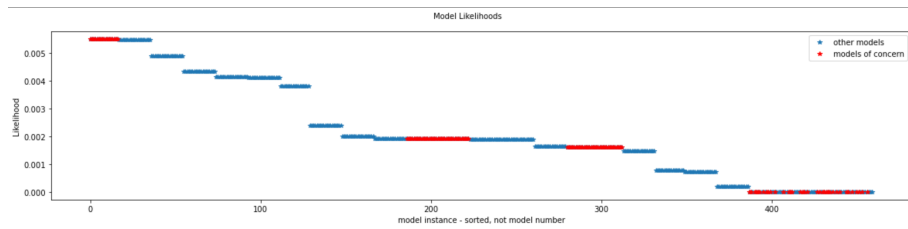
HW\_11.md

- Quinn Hull
- 04/12/2021
- HW11

'Correct' Figures:

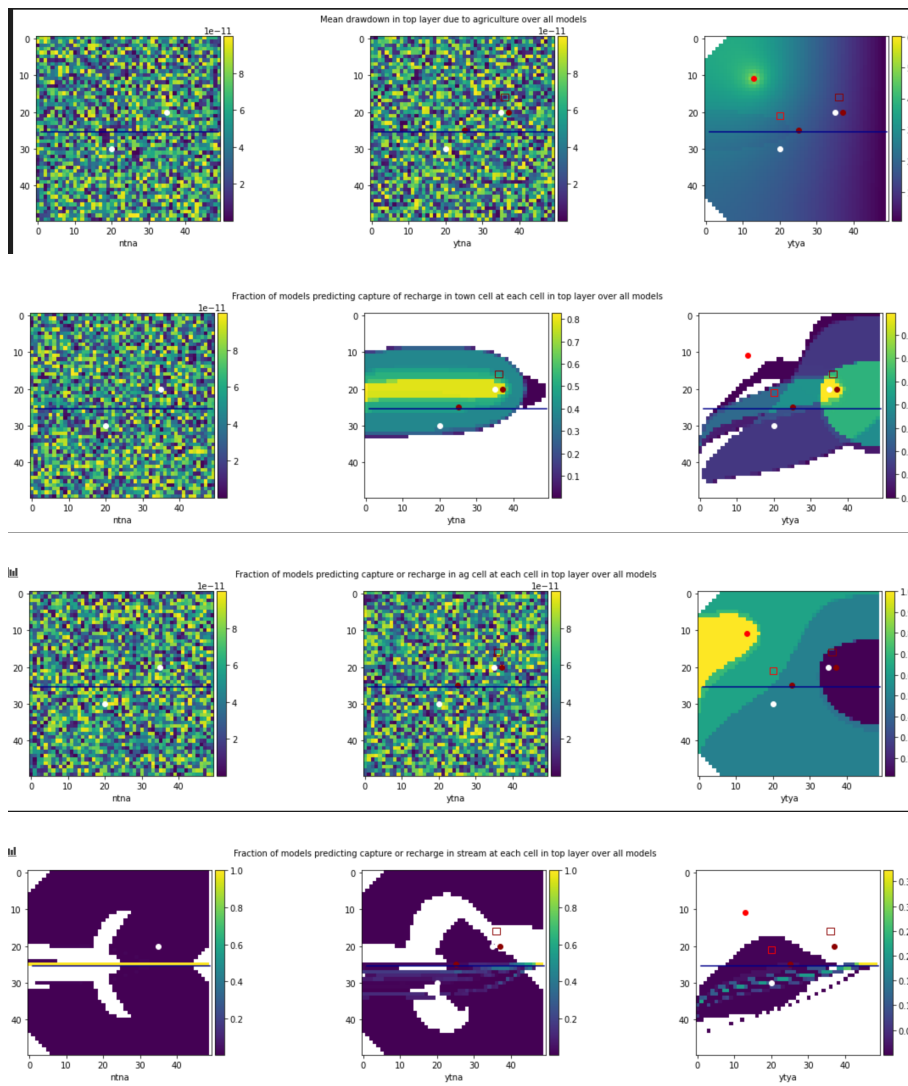
Ensemble 1 Probabilities





```
Models with highest likelihoods
m00100133114043 L = 0.005 model of concern
m001001332230232 L = 0.005 model of concern
m001001331440433 L = 0.005 model of concern
m001001333431141 L = 0.005 model of concern
m001001333430013 L = 0.005 model of concern
m001001332230232 L = 0.005 model of concern
m00100133114043 L = 0.005 model of concern
m001001333004112 L = 0.005 model of concern
m001001334240201 L = 0.005 model of concern
m00100133430013 L = 0.005 model of concern
The total likelihood of the models of concern is 0.217
```

## Ensemble 1 Figures



## Steps:

1. Get the run\_ensemble code running.
2. Empty the current model output directory.
3. Delete all of the m#### files from the output directory.
4. Run run\_ensemble.
5. Move the files from current output to output.
6. Run run\_ensemble and extract the results that you want.

7. Copy the m### files to a 'hold' directory in case you want that model set later.
8. Set up run\_ensemble to add models to your ensemble.
9. Repeat steps 2-8 to augment your ensemble.

## Context:

- Farm - proposal to add a well and irrigate crops
- Task: Use model to determine the risk of:
  - agrochemicals reaching the stream
  - reduced streamflow
  - additional drawdown in the town well
- Conditions:
  - No Town No Ag (NTNA)
  - Yes Town No Ag (YTNA)
  - Yes Town Yes Ag (YTYA)

## Challenge Steps:

1. Describe the scenario being modeled based on the fixed parameter values and the base model parameter values.
- Who is the stakeholder?
    - 'env'
  - What is their definition of an MOC?
    - MOC = 'Model of Concern' for an environmental stakeholder
      - moc\_time\_sequence = [2] <- YTYA
      - moc\_basis\_sequence = [2] <- Basis for determination of behavioral model, in this case 2 = streamflow at specified location
      - moc\_comparison\_sequence = [1] <- Less than
      - moc\_limit\_sequence = [50] <- Behavioral Criterion
      - moc\_column\_sequence = [38] <- Column of observation for point 2 or 3
      - moc\_row\_sequence = [25] <- Row of observation for point 2 or 3
  - What are the selected 'design' options of the ag facility and the town (return flow fraction, location, field location, etc)?
    - Ag:
      - farm\_nw\_row=[20, 34, 10, 20, 38] # north-western corner row of farm
      - farm\_nw\_col=[19, 19, 19, 10, 10] # north-western corner column of farm
      - irrig\_layer=[0, 0, 0, 0, 0] # layer to pump from for irrigation well
      - irrig\_row=[11, 18, 38, 31, 18] # irrigation well row
      - irrig\_col=[13, 30, 13, 30, 42] # irrigation well col
      - farm\_landuse = 0.125 # fraction of total farm area in active use at any time
      - farm\_efficiency = 0.7 # fraction of pumped water actually delivered \*farm\_excess = 0.2 # fraction of crop demand to be added to prevent salinization
      - crop\_demands = [0.004, 0.006, 0.008] # water use (m/day) for each crop [wheat, pistachios, cotton]
    - Town:
      - return\_column=[10,15,20,25,30] # column at which town return flow is added to stream
      - rech\_layer=[0, 0, 0, 0, 0] # recharge basin layer
      - rech\_nw\_row=[29, 20, 15, 15, 30] # recharge basin north-western corner row
      - rech\_nw\_col=[19, 29, 5, 35, 40] # recharge basin north-western corner column

- Qw10 = 1000. # initial pump rate when well 1 is turned on at beginning of post development period
- rate = 0.0405 # exponential growth rate of town's pumping
- town\_efficiency = 0.8 # fraction of pumped water actually delivered to town
- town\_consumption = 0.5 # fraction of delivered water that is consumed, not reclaimed
- well1\_loc=[2,20,37] # location of town well
- town\_recharge\_ratio = [0,.25, .5, .75, .9] # fraction of town's reclaimed water recharged vs. returned to stream (-)

2. Construct an ensemble with 25 unique parameter sets chosen at random and generate output in current model output.

```

# Import the random module
import random

# Create a list of parameter sets
param_sets = [
    {'Qw10': 1000, 'rate': 0.0405, 'town_efficiency': 0.8, 'town_consumption': 0.5, 'well1_loc': [2, 20, 37], 'town_recharge_ratio': [0, 0.25, 0.5, 0.75, 0.9]},
    # ... (24 more unique parameter sets)
]

# Generate 25 unique parameter sets
for i in range(25):
    param_sets.append({
        'Qw10': random.uniform(1000, 1000),
        'rate': random.uniform(0.0405, 0.0405),
        'town_efficiency': random.uniform(0.8, 0.8),
        'town_consumption': random.uniform(0.5, 0.5),
        'well1_loc': random.choice([2, 20, 37]),
        'town_recharge_ratio': random.choice([0, 0.25, 0.5, 0.75, 0.9])
    })

```

- I attempted to follow instructions outlined in tutorial, namely:

```

there is a total of 25 run(s)

['m00100133244041', 'm001001332100033', 'm001001331034104',
 'm001001331000120', 'm001001332000111', 'm001001334201130',
 'm001001332230252', 'm001001334311244', 'm001001330243421',
 'm001001332130433', 'm001001330310110', 'm001001332440002',
 'm001001332144042', 'm001001334031114', 'm001001330221100',
 'm001001333413130', 'm001001331144043', 'm001001334044223',
 'm001001331101343', 'm001001333430013', 'm001001334240201',
 'm001001333004112', 'm001001331243410', 'm001001333130143',
 'm001001333431141']

```

- Which created in `..._run_class_ensemble.ipynb` 25 models

3. Remove all of the `m###` models from output and move all of the models from current model output to output.

- Okay

4. Run `analyze_ensemble` and construct your version of the Key Figures - Ensemble 1.

```

The final ensemble includes 460 models
There were 73 non-behavioral models.

```

- When I do this and run the analyzer, I find that I have 460 models!
- See figures

5. Use the results to identify one MOC and use that to generate 10 additional similar models.

- @Ty How do I do this?

6. Move the `m###` model results from current model output to output.

- Incomplete

7. Rerun `analyze_ensemble` and construct your version of the Key Figures - Ensemble 2.

- Incomplete

## Challenge Questions

1. Based on your initial random ensemble, what is the most likely additional drawdown at the town well due to pumping the ag well? How confident are you in that response - explain/defend your answer.
2. What is the likelihood that the reality (represented by the meager observed data) is best represented by an MOC?
3. What is the most likely loss in streamflow at the outflow end of the domain? Justify your answer.
4. Is it likely that either the town or ag well could be contaminated by the ag field? Justify your answer.
5. Make a set of plots based on ensemble 2 and discuss how each of your answers to the first four questions changed due to adding the MOC-inspired parameter sets.