Gaussian Process Emulation with Agent Based Models: A Worked Example

Louise Kimpton
Peter Challenor
James Salter

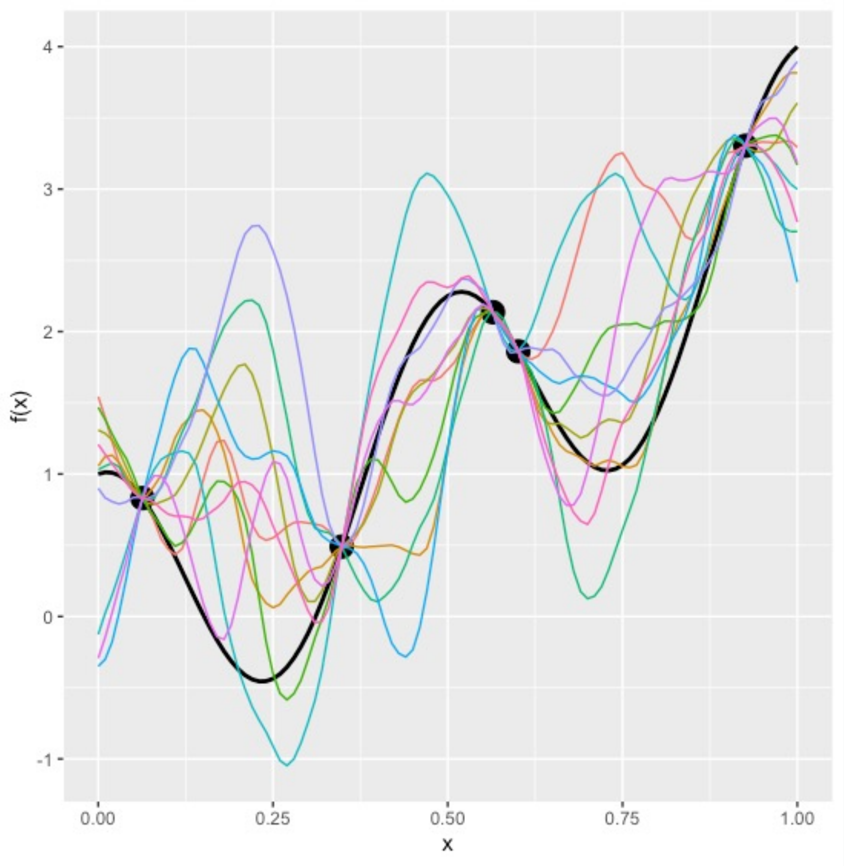
Email: l.m.kimpton@exeter.ac.uk

Introduction

- Uncertainty quantification for Agent Based Models
- Methods applied to a test model: wolf and sheep predator-prey model on NetLogo
- Model has been simplified to have 2 inputs and 1 output to be easily presented, but methods can be scaled to much more difficult problems

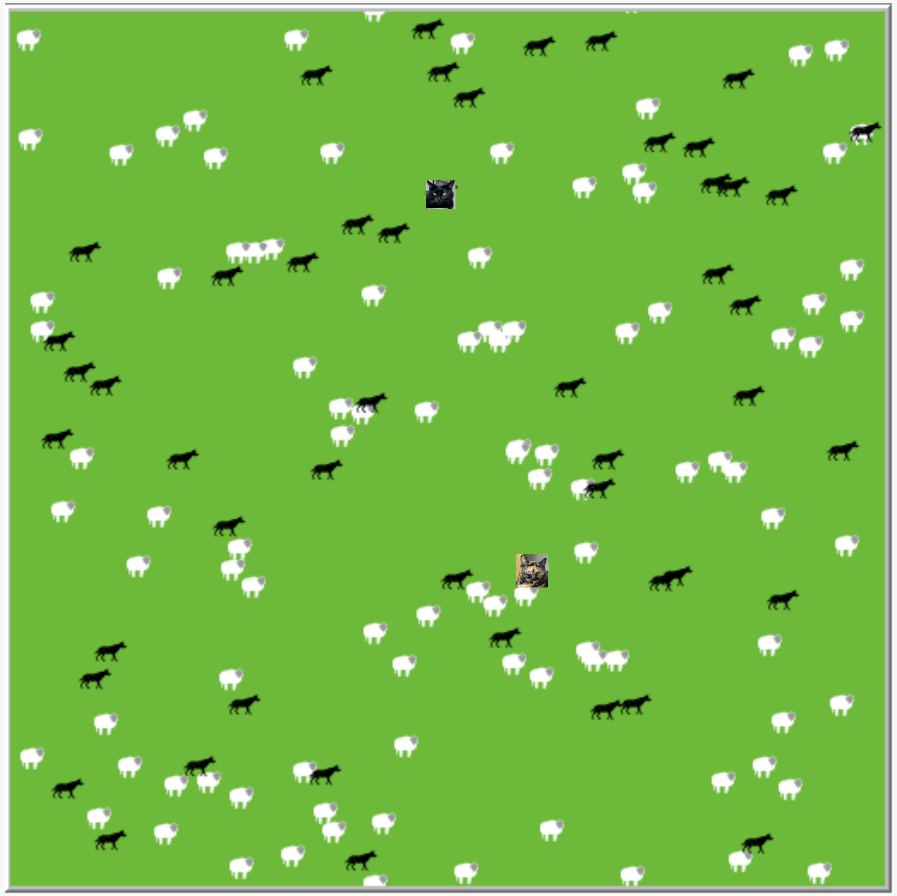
UQ Methods

- Design
- Classification
- Emulation – Gaussian process
- Stochastic GPs
- Sequential design
- Stochastic design
- Calibration – history matching



Wolf Sheep Predation Model

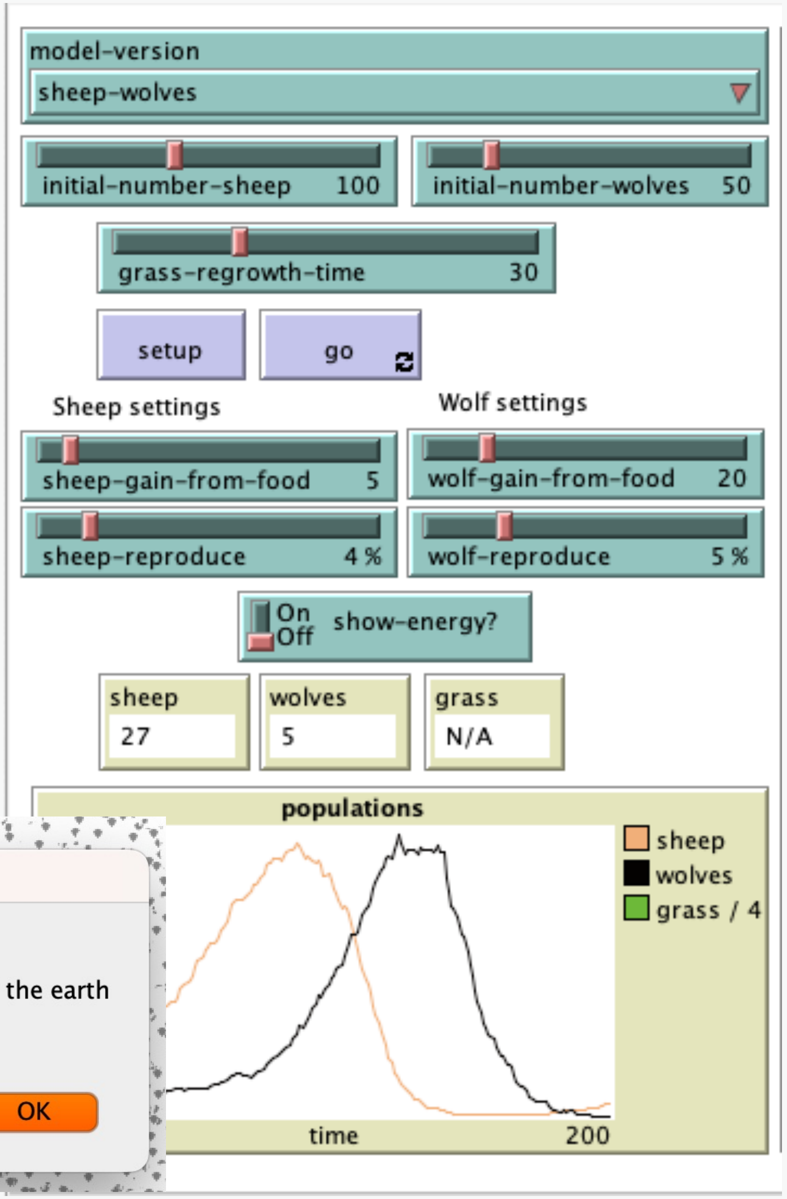
- Wolf and sheep predator-prey ecosystem ABM model
- Sheep wander randomly, wolves look for sheep to eat
- Each step costs wolves energy, they must eat sheep to replace energy – when they run out of energy, they die
- Each wolf and sheep has a fixed probability of reproduction
- Grass is infinite – eating or moving doesn’t change the sheep’s energy levels
- The system is unstable if it results in extinction for at least one species



Wolf Sheep Prediction Model

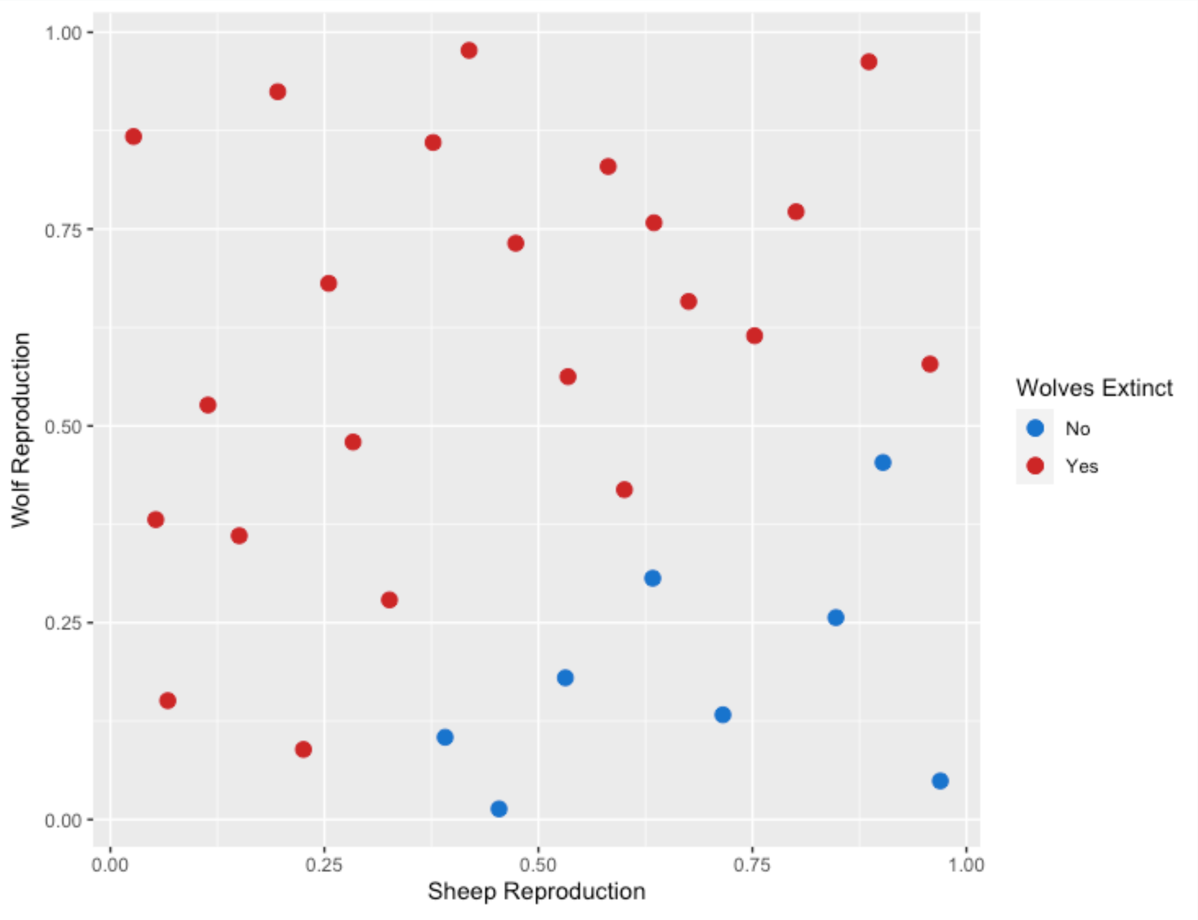
- Inputs to the ABM:
  - 2d problem so easily visualised
  - Sheep reproduction % $\in [0, 20]$
  - Wolf reproduction % $\in [0, 20]$
  - Fix all other parameters at default

- Outputs to the ABM:
  - Model records numbers of sheep and wolves at each time step
  - Stochastic output
  - Three scenarios:
    - Sheep die out first, then wolves die out
    - Wolves die out first, then sheep inherit the earth
    - Neither die out
  - Choose the output to be the time to wolf extinction



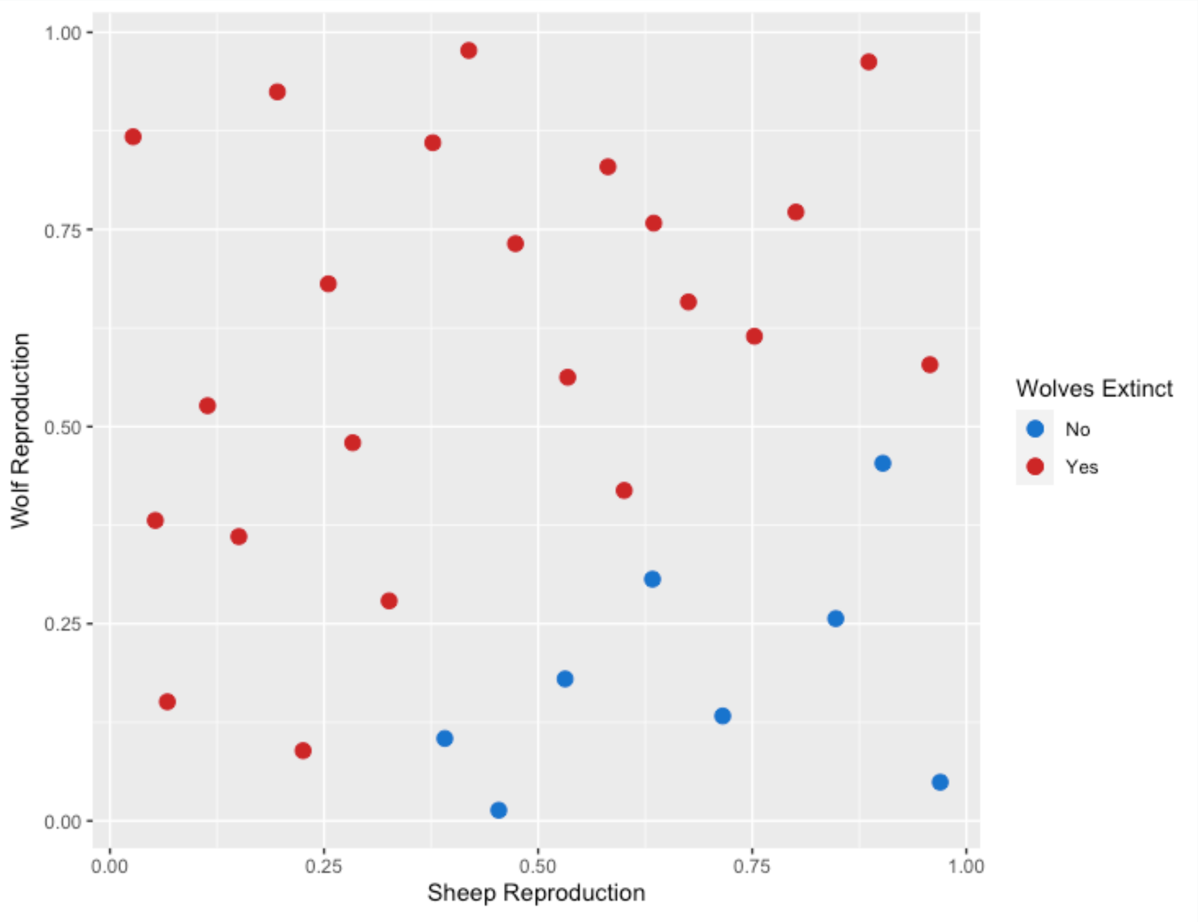
Design

- Where are we first going to run the model?
- Aim to be space filling given a limited budget of runs
- Choose a design using a maximin Latin hypercube with 30 points
- Since the model is stochastic, run the Latin hypercube at 10 replicates
- An area of input space (blue points) has no output - wolves don't go extinct



Classification

- Can we predict where the wolves will/won't go extinct?
- Map the area with no output: know where not to put future runs
- Or can place more points near boundary to improve classification prediction
- Methods include logistic regression and GP classification

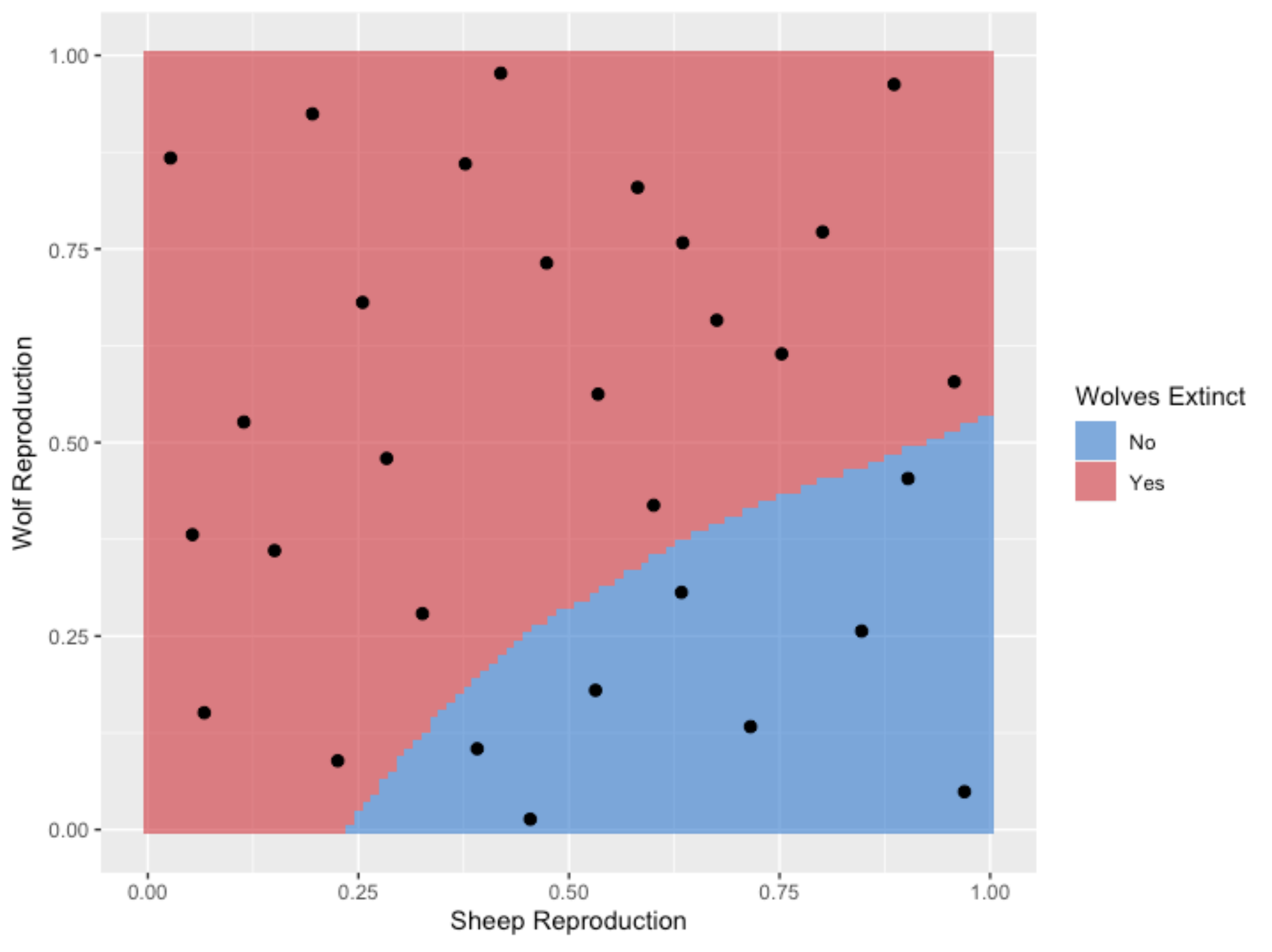


Classification

- Each input x is given a class label corresponding to its given region:
  - l_1 if $x \in wolves\ extinct = yes$
  - l_2 if $x \in wolves\ extinct = no$

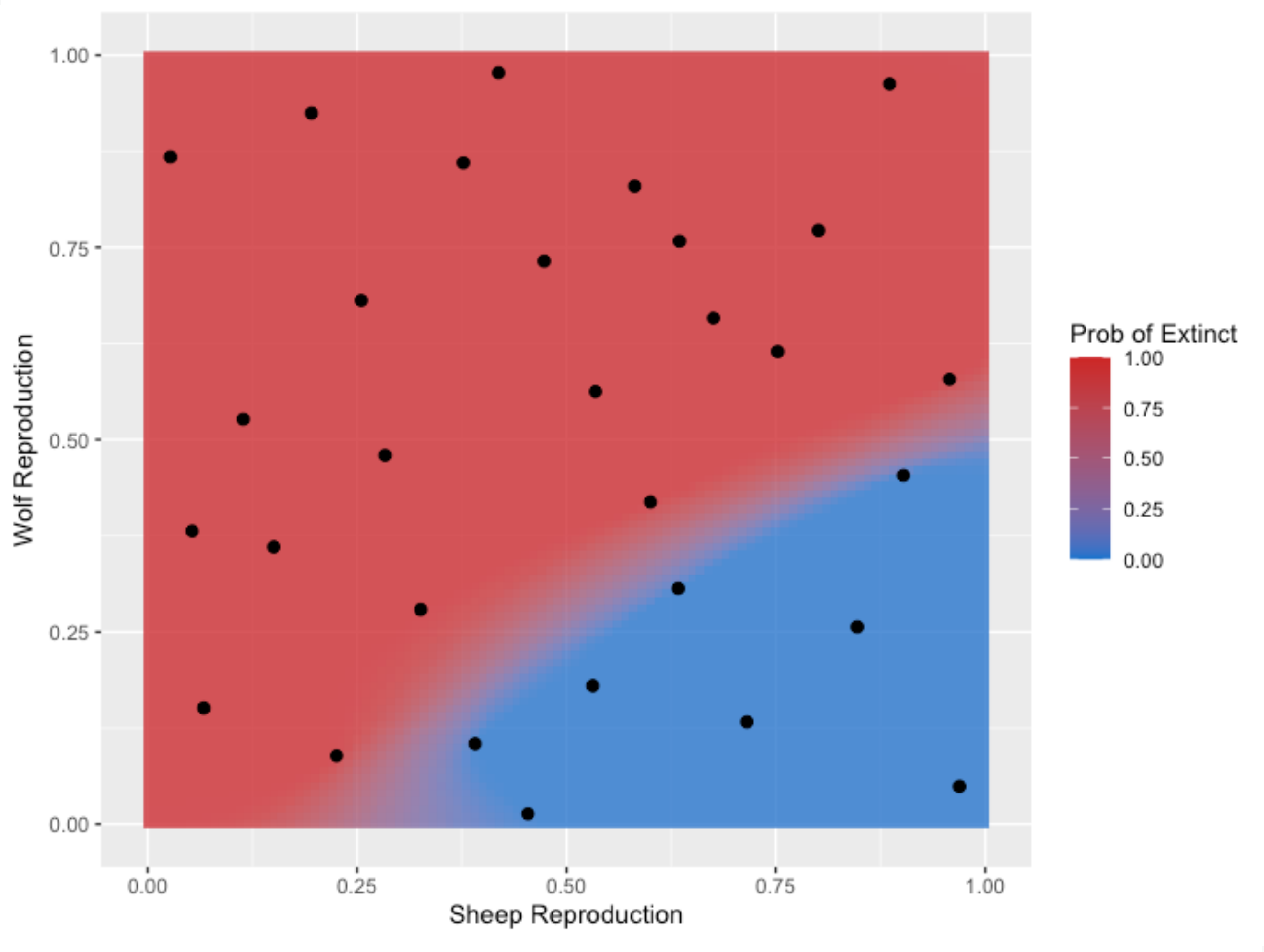
- Build a latent Gaussian process over the boundary labels conditioned on the GP being:
  - Negative at points labelled l_1
  - Positive at points labelled l_2

- Use GP to predict across the full input space thresholding at zero



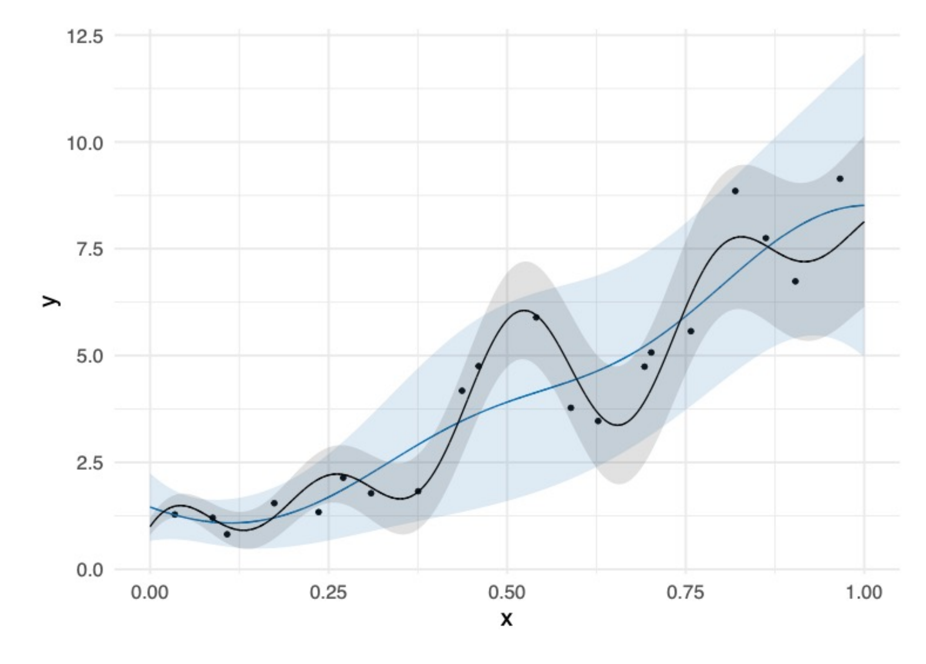
Classification

- Each input x is given a class label corresponding to its given region:
  - l_1 if $x \in wolves\ extinct = yes$
  - l_2 if $x \in wolves\ extinct = no$
- Build a latent Gaussian process over the boundary labels conditioned on the GP being:
  - Negative at points labelled l_1
  - Positive at points labelled l_2
- Use GP to predict across the full input space thresholding at zero



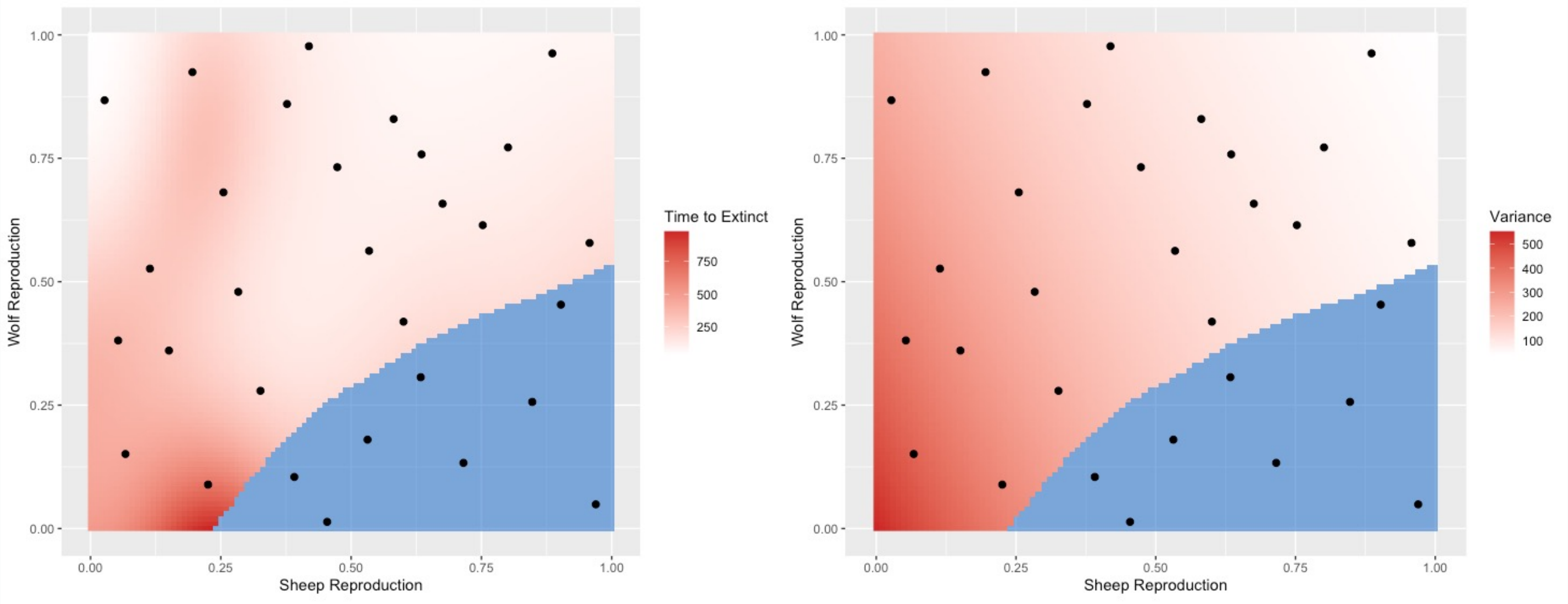
Stochastic Emulation

- Most ABMs are stochastic – if we run the model at the same input parameters multiple times, we get different outputs
- Deterministic GPs are no longer suitable – the variance can depend on the input
- Stochastic Gaussian processes:
  - If there are enough replicates at each input: fit two GPs, one to the sample means and one to the sample variances
  - If there are not enough replicates at each input: alternative methods including hetGP (Binois and Gramacy, 2018)



Reference: Binois, M., Gramacy, R. B., and Ludkovski, M. (2018). Practical heteroscedastic gaussian process modelling for large simulation experiments. Journal of Computational and Graphical Statistics

Stochastic Emulation

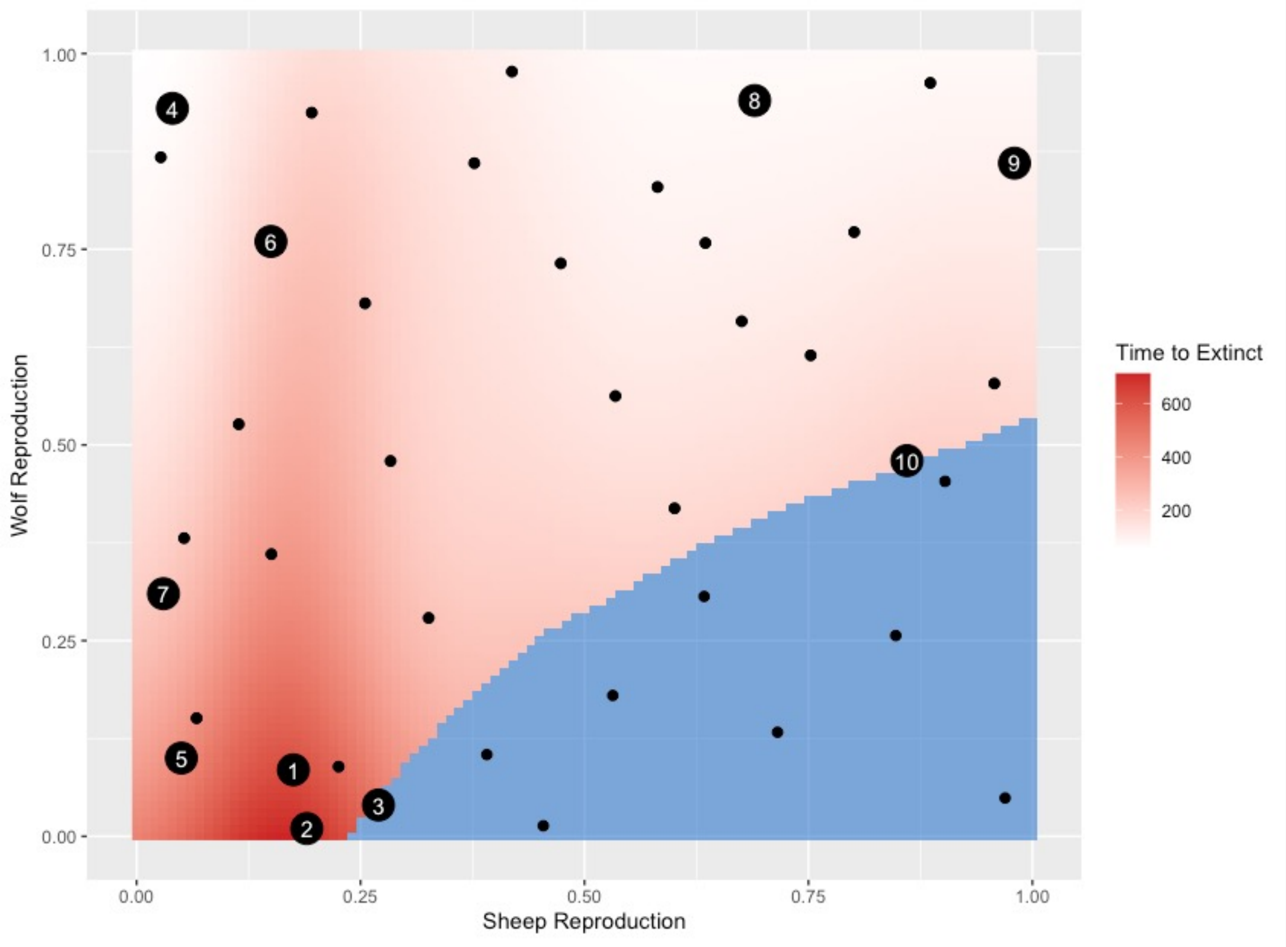


Sequential Design

- Choose the next best design point to run the ABM model at to maximally improve the fit of the GP
- Choice of input point is typically based on a trade-off between exploration and exploitation
- Sequential designs differ to one-shot designs – we choose one design point at a time to update our current GP
- For stochastic models we can focus on just improving the mean estimate, or to simultaneously improve both the mean and variance

Deterministic Sequential Design

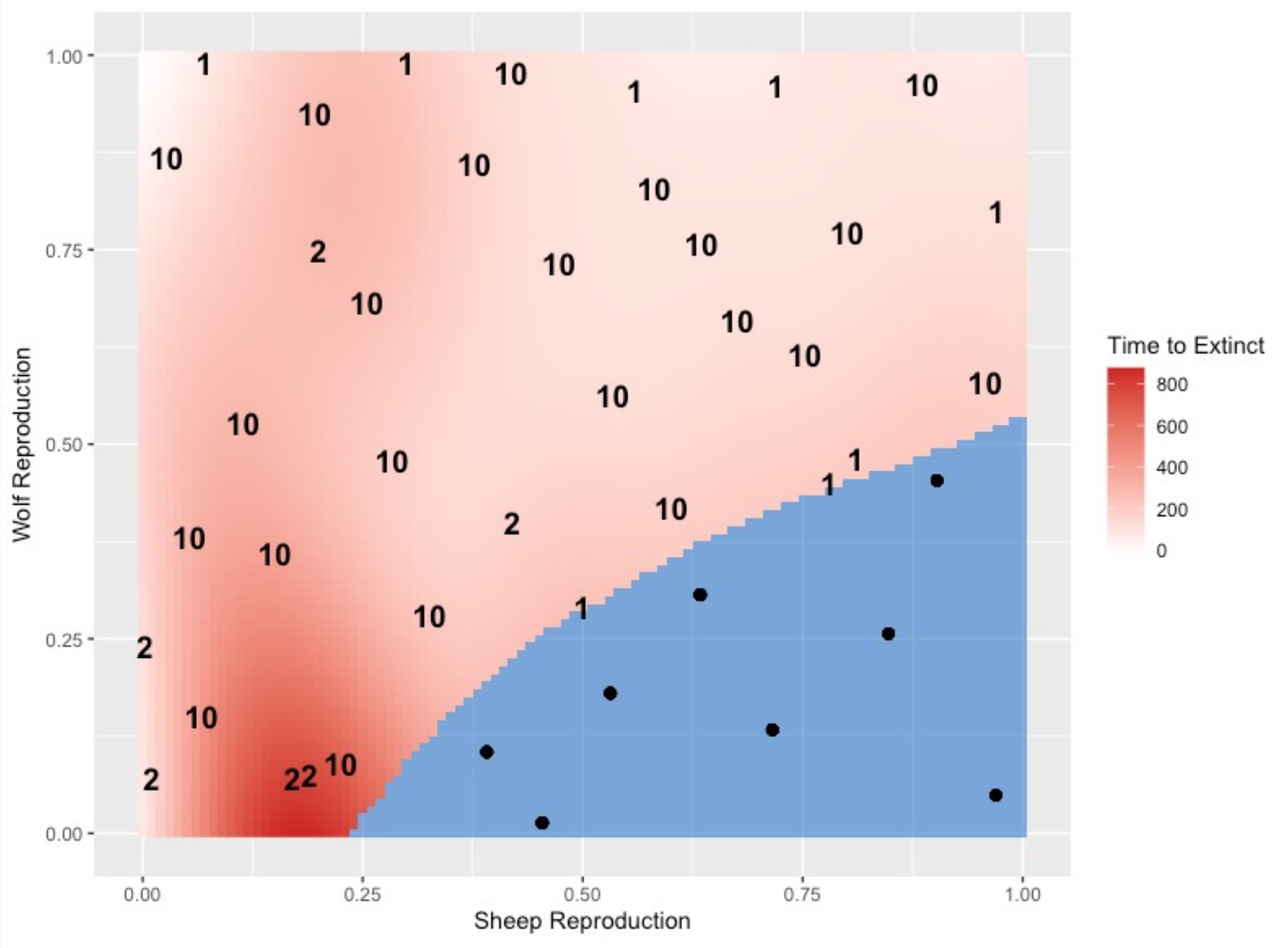
- Aim to include new runs of the model to improve the fit of the sample-mean GP
- Based on calculating the expected squared leave-one-out error at each of the design points
- The next point is then chosen using pseudo expected improvement (PEI): a modified expected improvement with repulsion function
- At each chosen location, we run the ABM at 10 replicates and update our GP



Reference: Cross-Validation Based Adaptive Sampling for Gaussian Process Models, (2022), H. Mohammadi, P. Challenor, D. Williamson, M. Goodfellow, SIAM/ASA Journal on Uncertainty Quantification

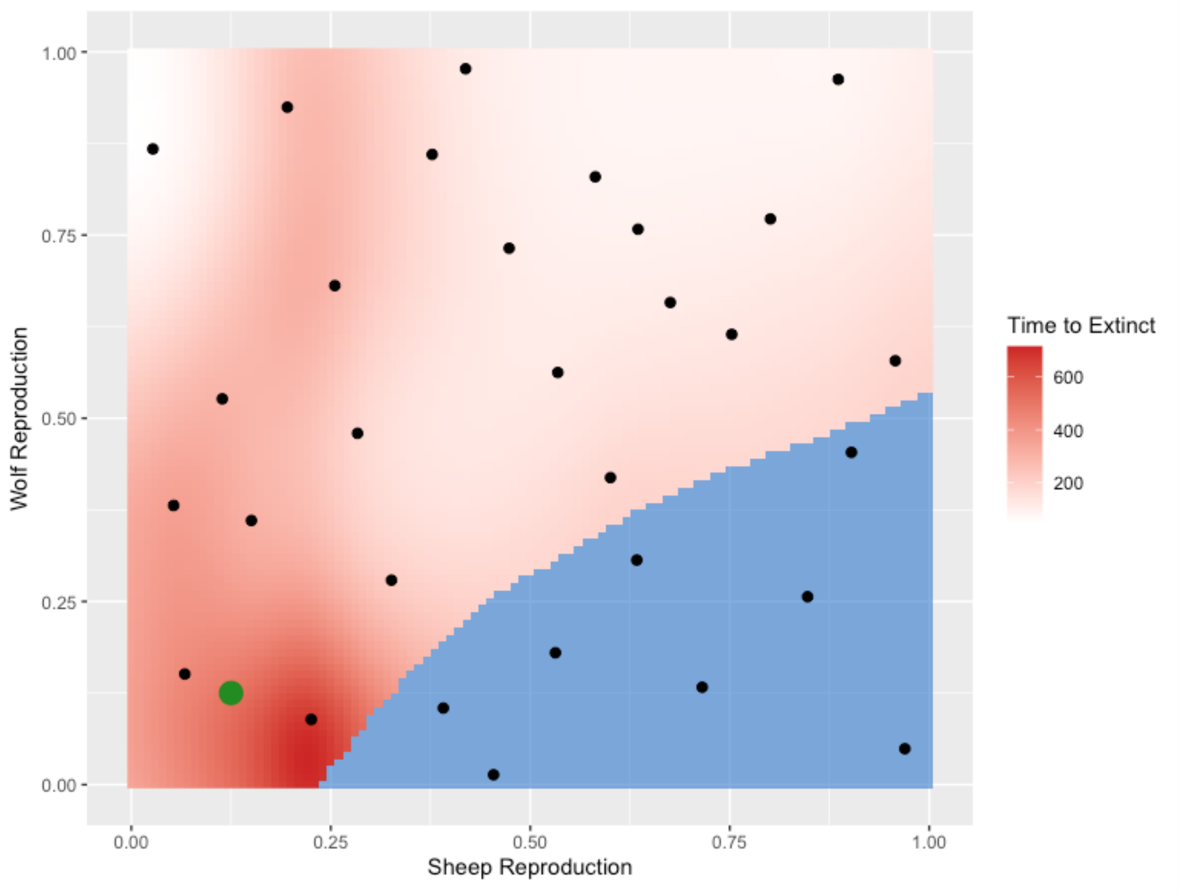
Stochastic Sequential Design

- Aim to include new runs of the model to improve the fit of the GP for BOTH the mean and variance
- At each iteration there is a choice to include either a new point OR a replicate at an existing point
- Based on a similar method to the pseudo expected improvement (PEI) extended to stochastic GPs using hetGP
- Cheaper and more efficient



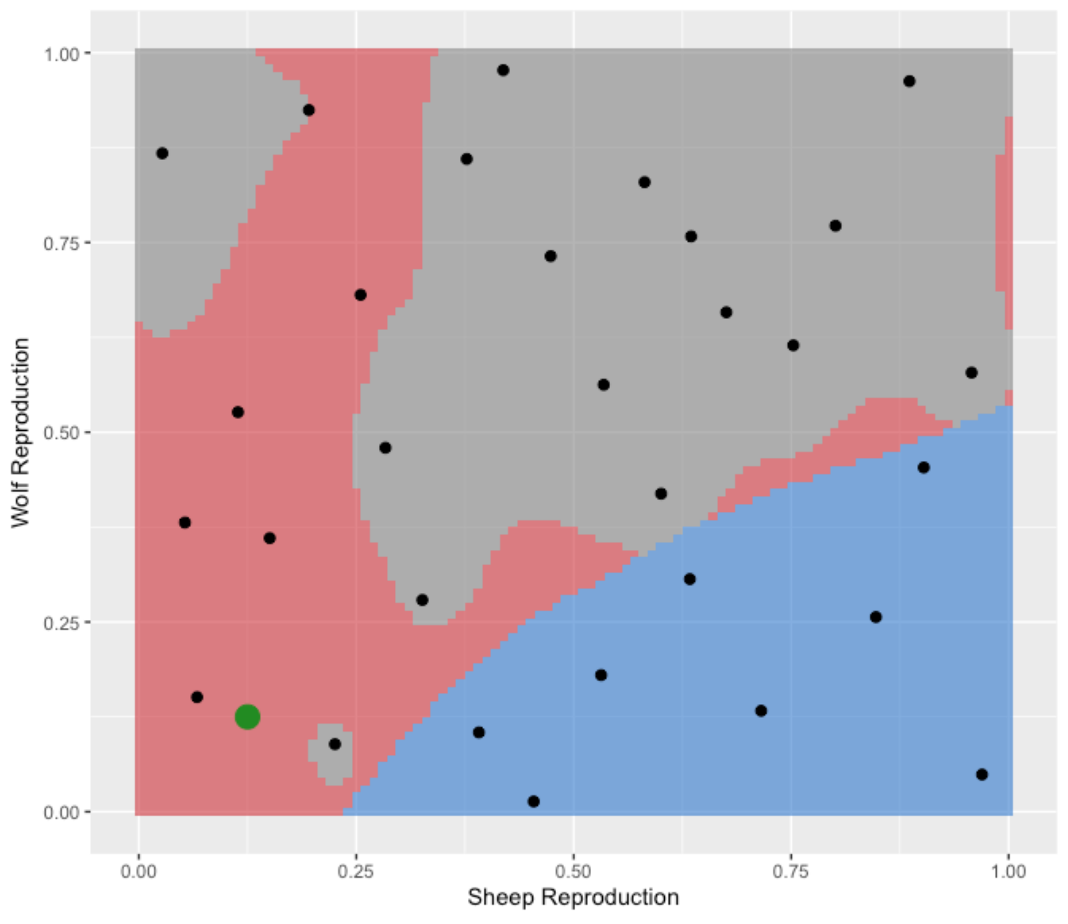
Calibration: history matching

- Given an observation, can we use our emulator to estimate the most likely input parameters?
- History matching rules out regions of parameter space that are not consistent with the observation using an implausibility metric
- The implausibility is based on the distance between the observation and the model prediction (given some error)
- If the distance is large, the points are ruled out, otherwise they are 'not ruled out yet'

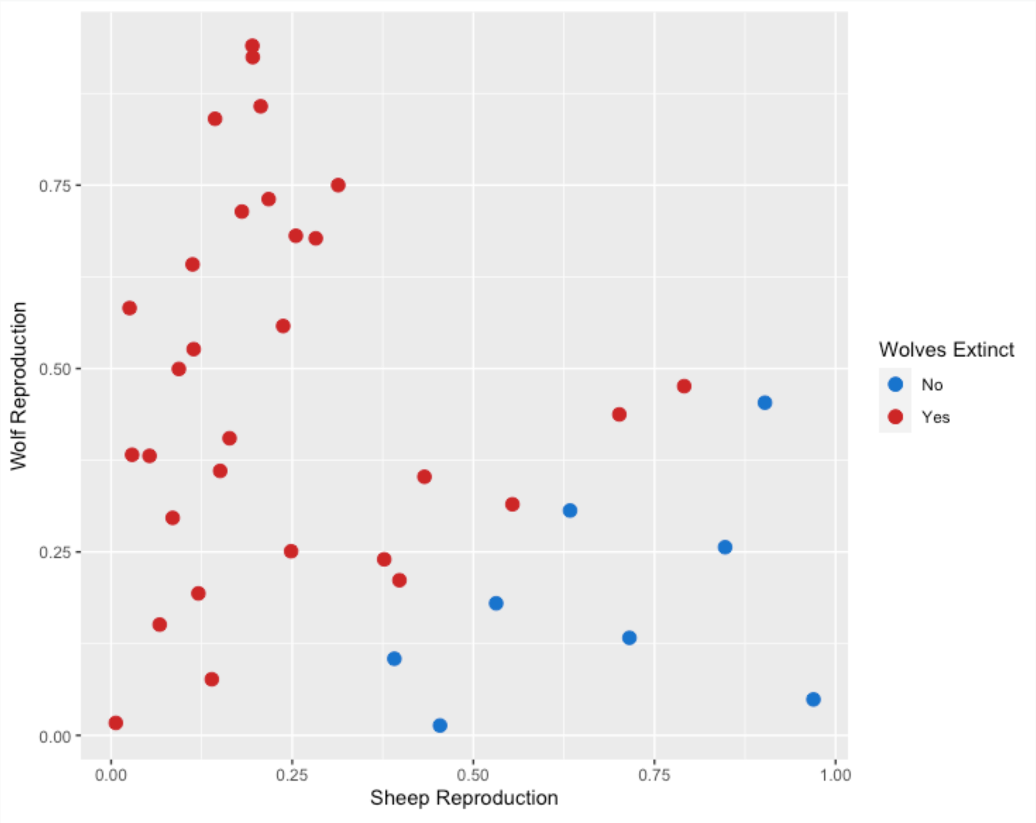


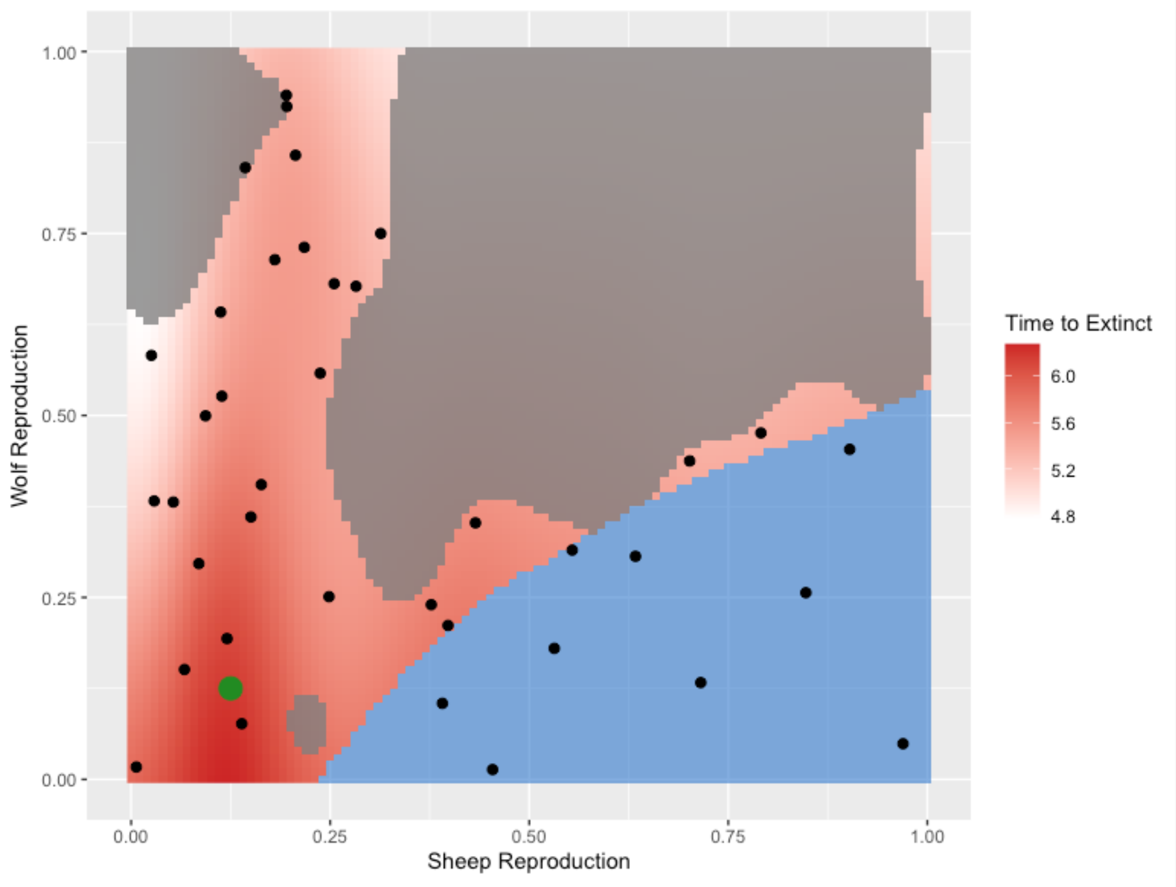
Calibration: history matching

- History matching is performed in waves to focus in on good regions of input space:
    1. Set up an initial input design
    2. Run design through model
    3. Build Gaussian process
    4. History match to rule out space
    5. Sample new points from NROY space
    6. Run new points through model
    7. Build new Gaussian process
    8. History match to rule out space
    9. …

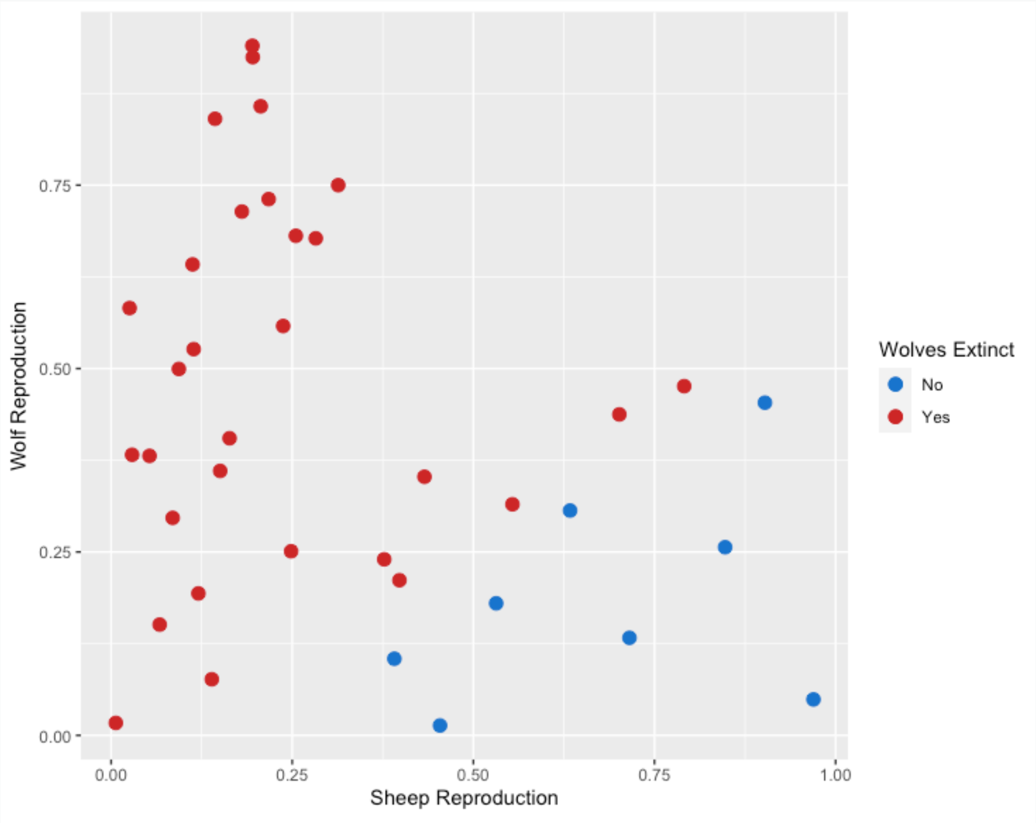


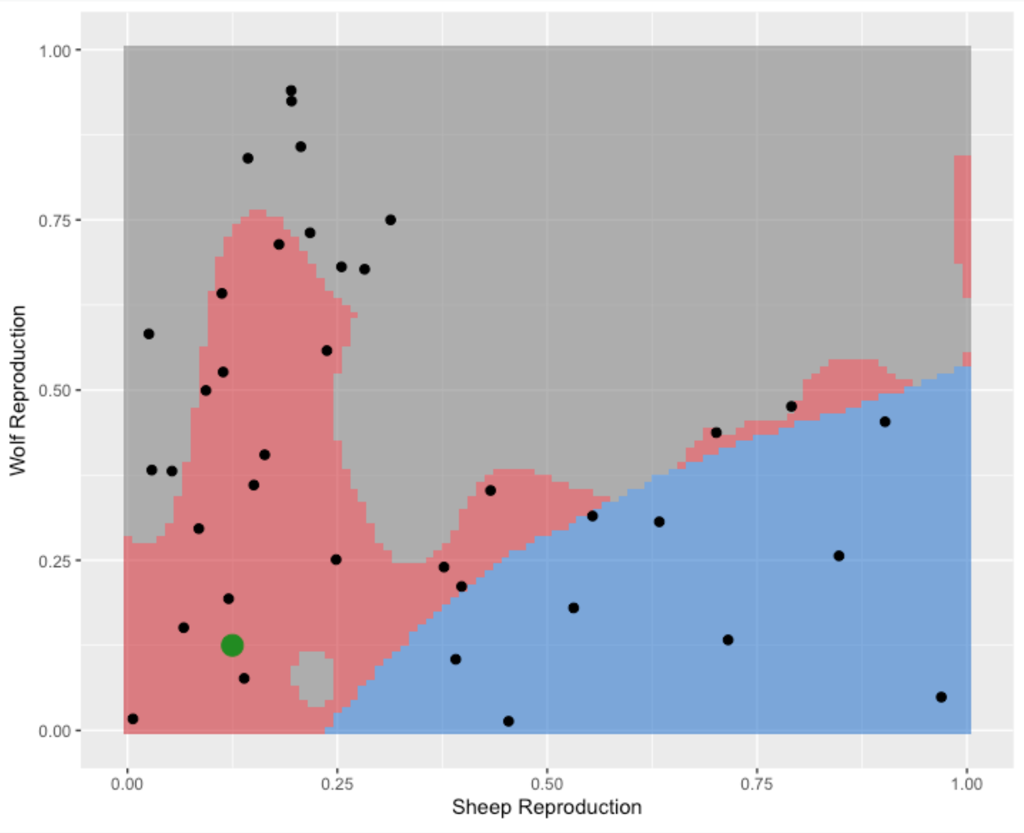
Calibration: history matching



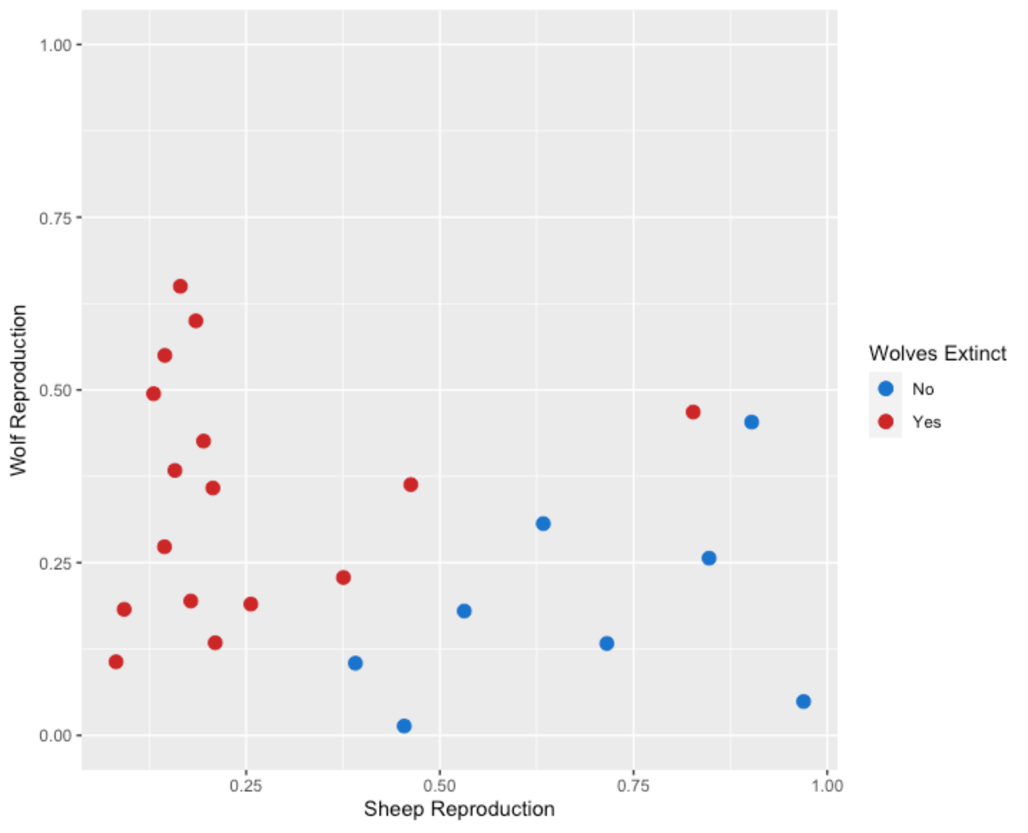


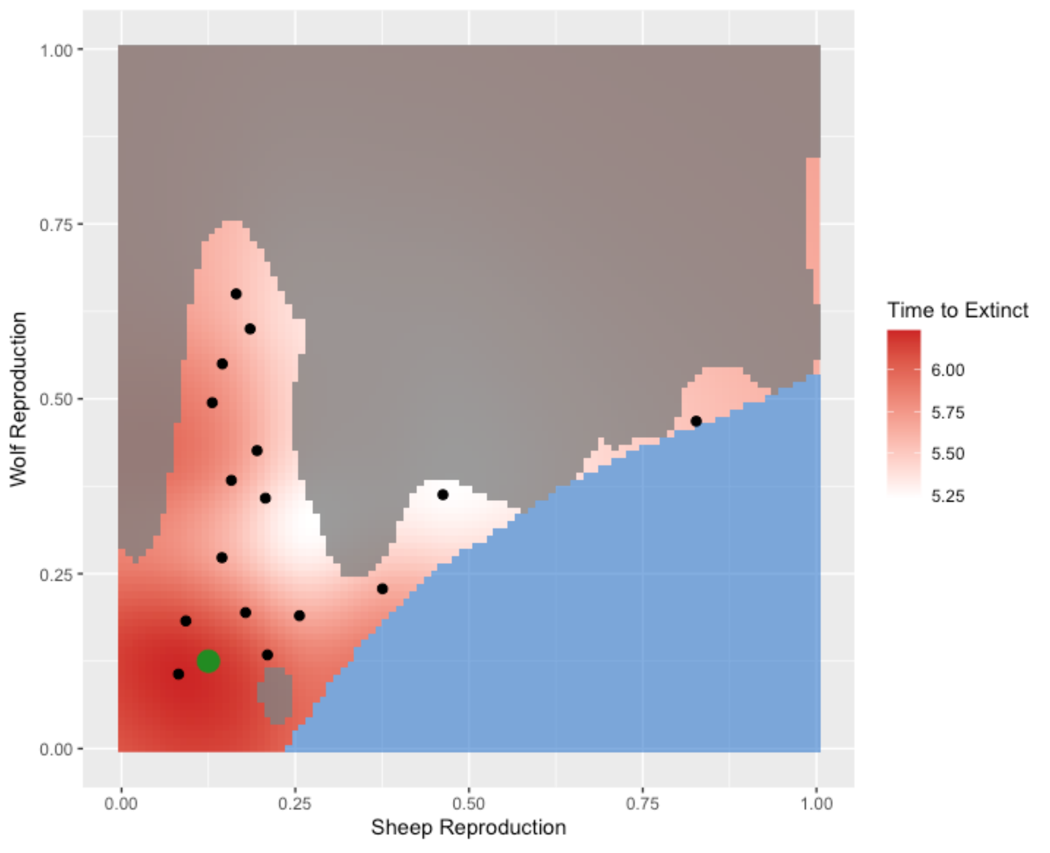
Calibration: history matching



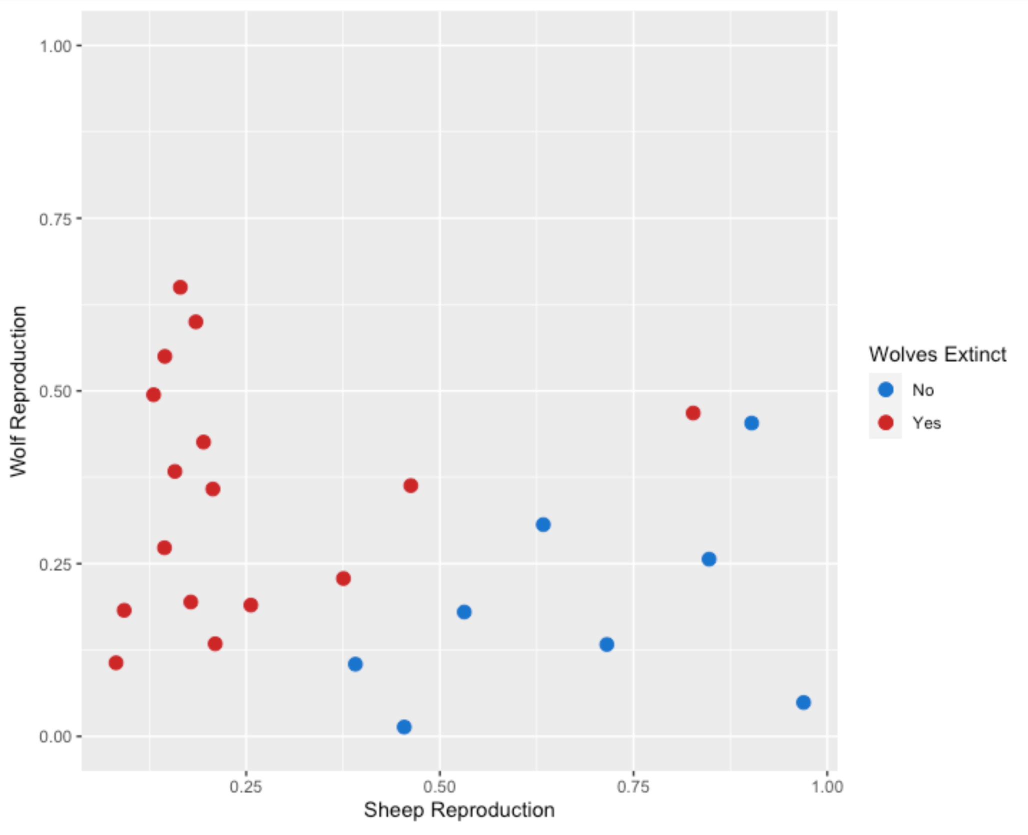


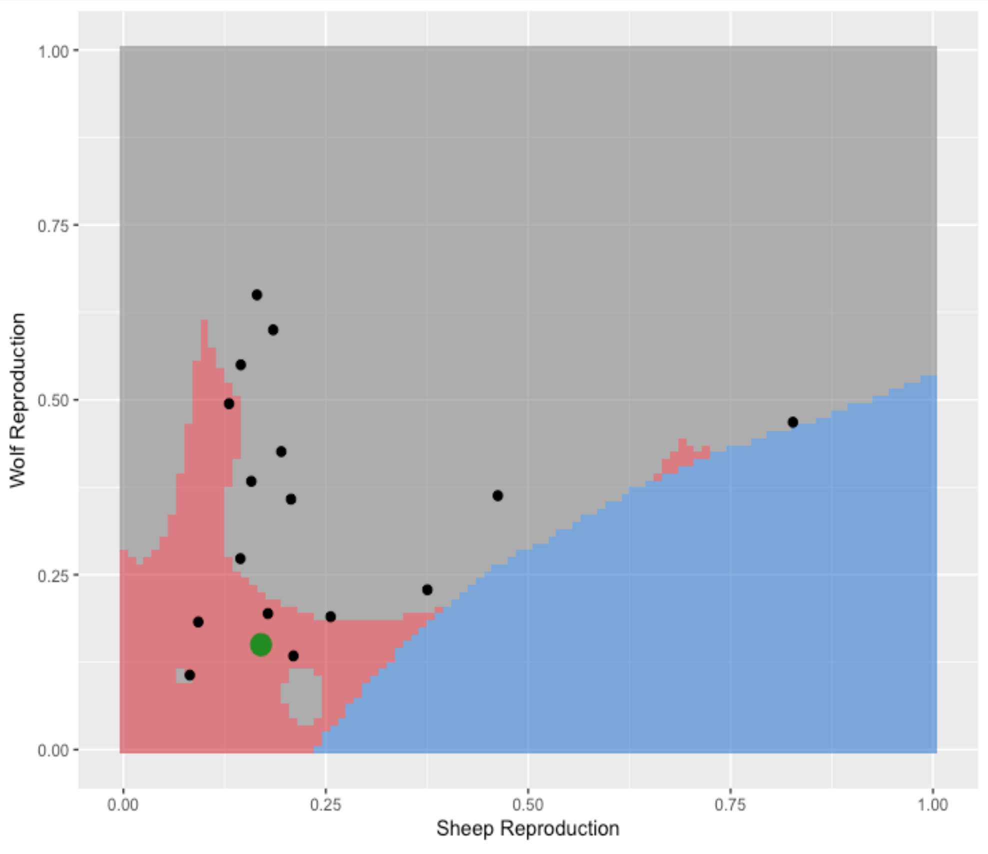
Calibration: history matching





Calibration: history matching





Conclusion

- Uncertainty quantification for Agent Based Models
- Methods include emulation, classification, sequential design and history matching
- Methods applied to wolf and sheep predator-prey model but can be applied to many other models