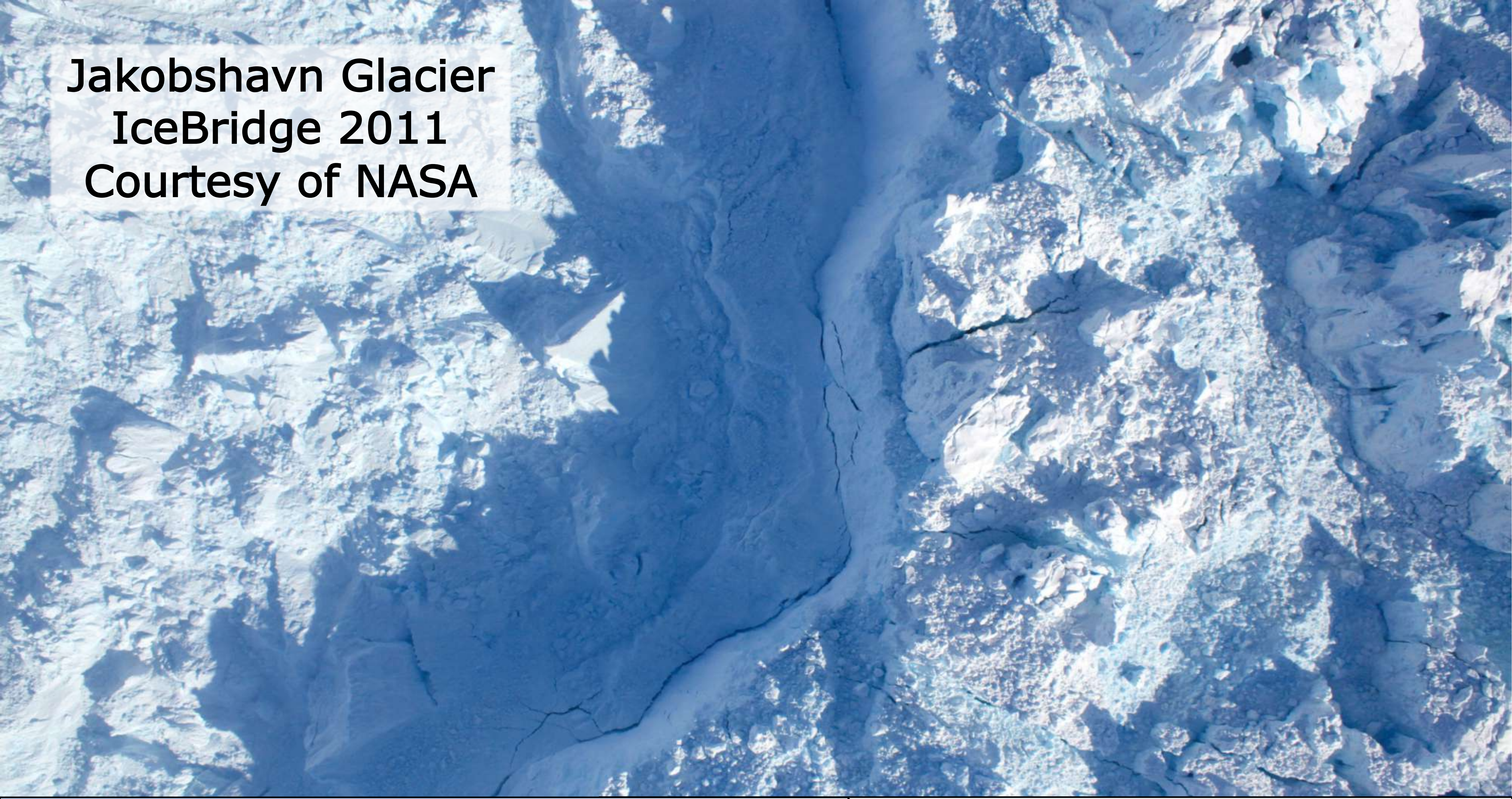
Python as a Self-Teaching Tool: Insights into Gaussian Process Modeling using Python Packages

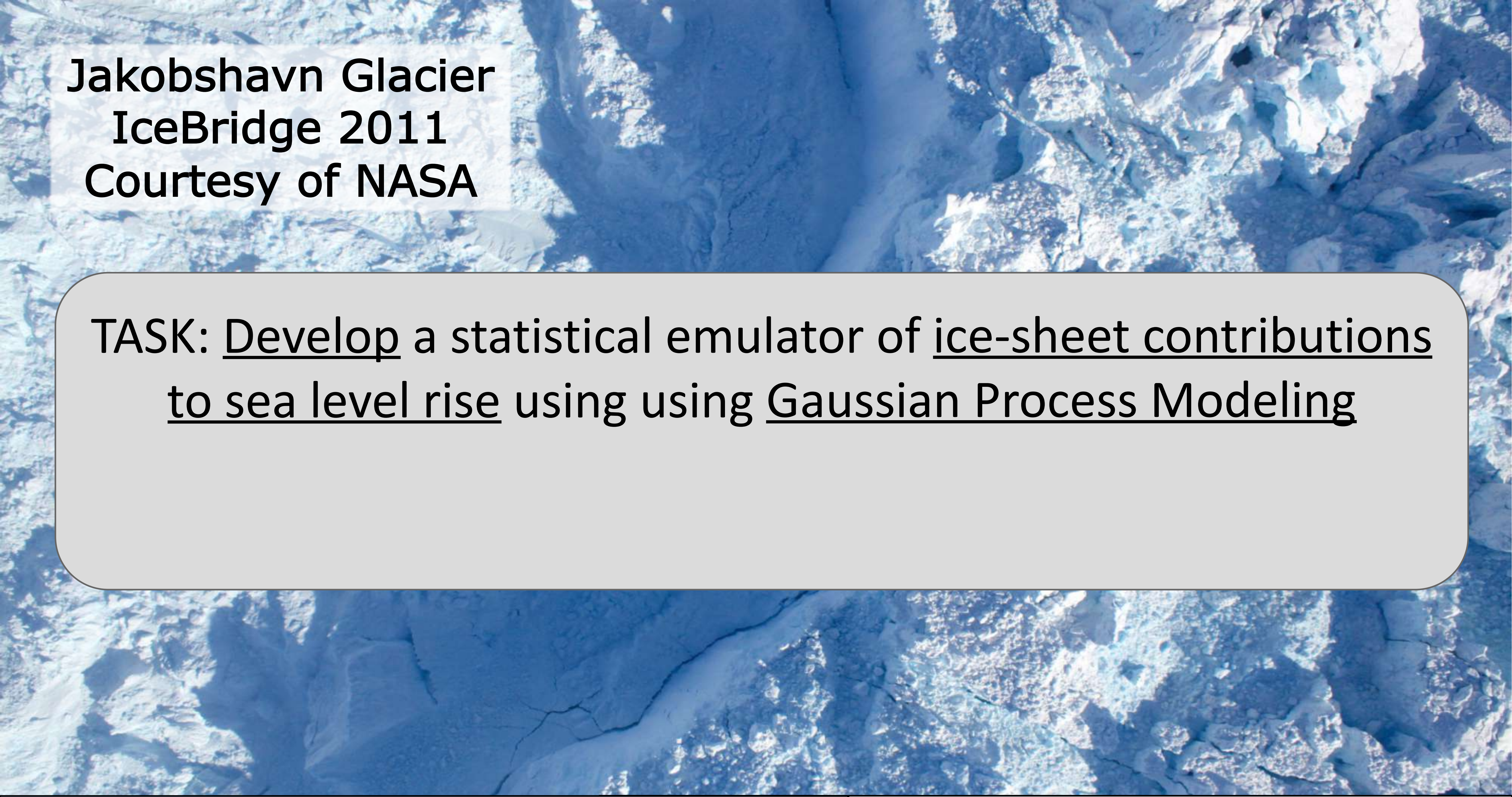
Daniel Gilford daniel.gilford@rutgers.edu @danielgilford

Collaborators: Robert Kopp, Erica Ashe, Rob DeConto, David Pollard, Anna Ruth Halberstadt, Ian Bolliger, Michael Delgado, Moon Limb

Rutgers Institute of Earth, Ocean, and Atmospheric Sciences

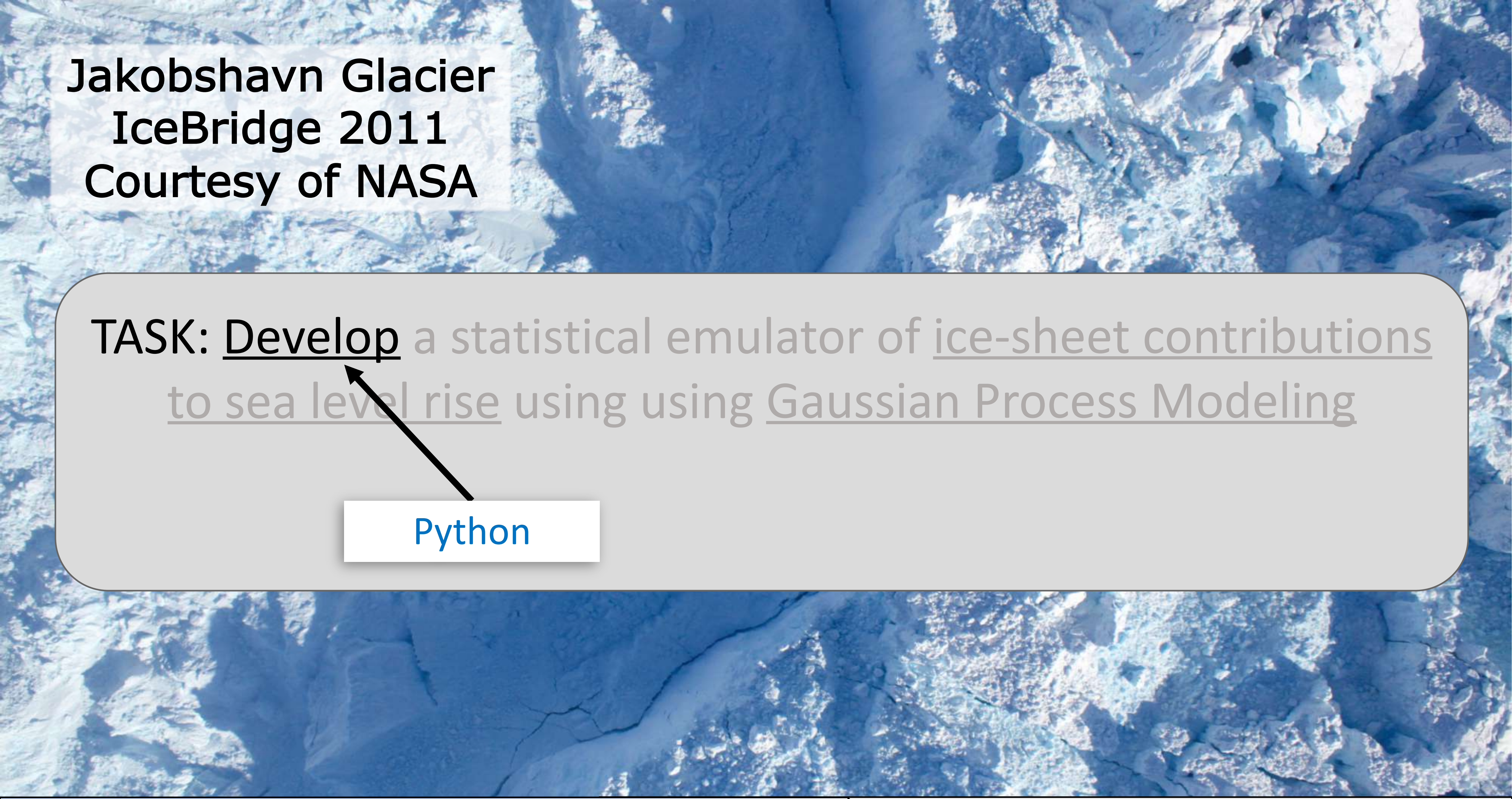
Support From: NSF NASA


Jakobshavn Glacier
IceBridge 2011
Courtesy of NASA



Jakobshavn Glacier
IceBridge 2011
Courtesy of NASA

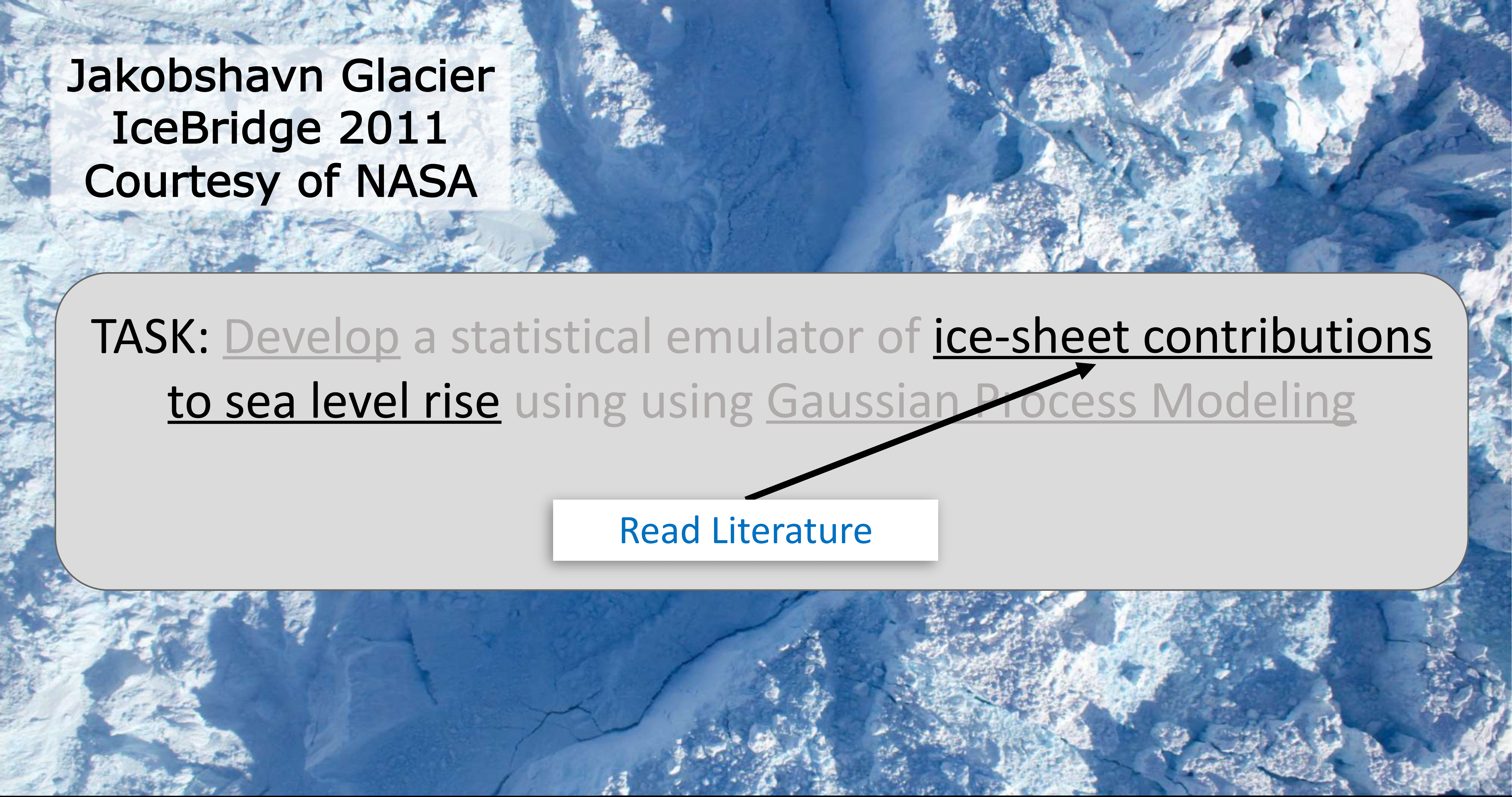
TASK: Develop a statistical emulator of ice-sheet contributions to sea level rise using using Gaussian Process Modeling



Jakobshavn Glacier
IceBridge 2011
Courtesy of NASA

TASK: Develop a statistical emulator of ice-sheet contributions to sea level rise using using Gaussian Process Modeling

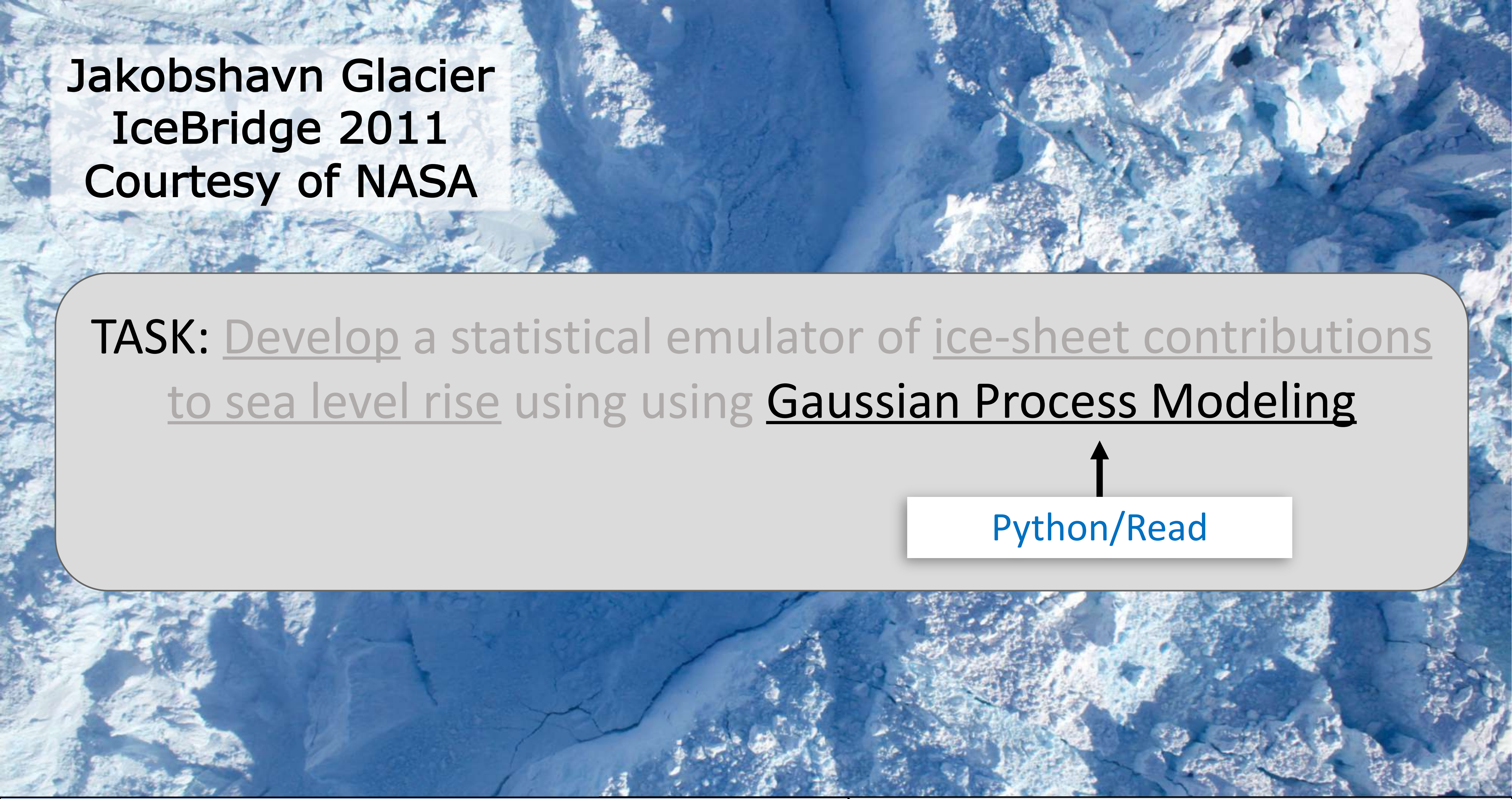
Python



Jakobshavn Glacier
IceBridge 2011
Courtesy of NASA

TASK: Develop a statistical emulator of ice-sheet contributions to sea level rise using using Gaussian Process Modeling

Read Literature

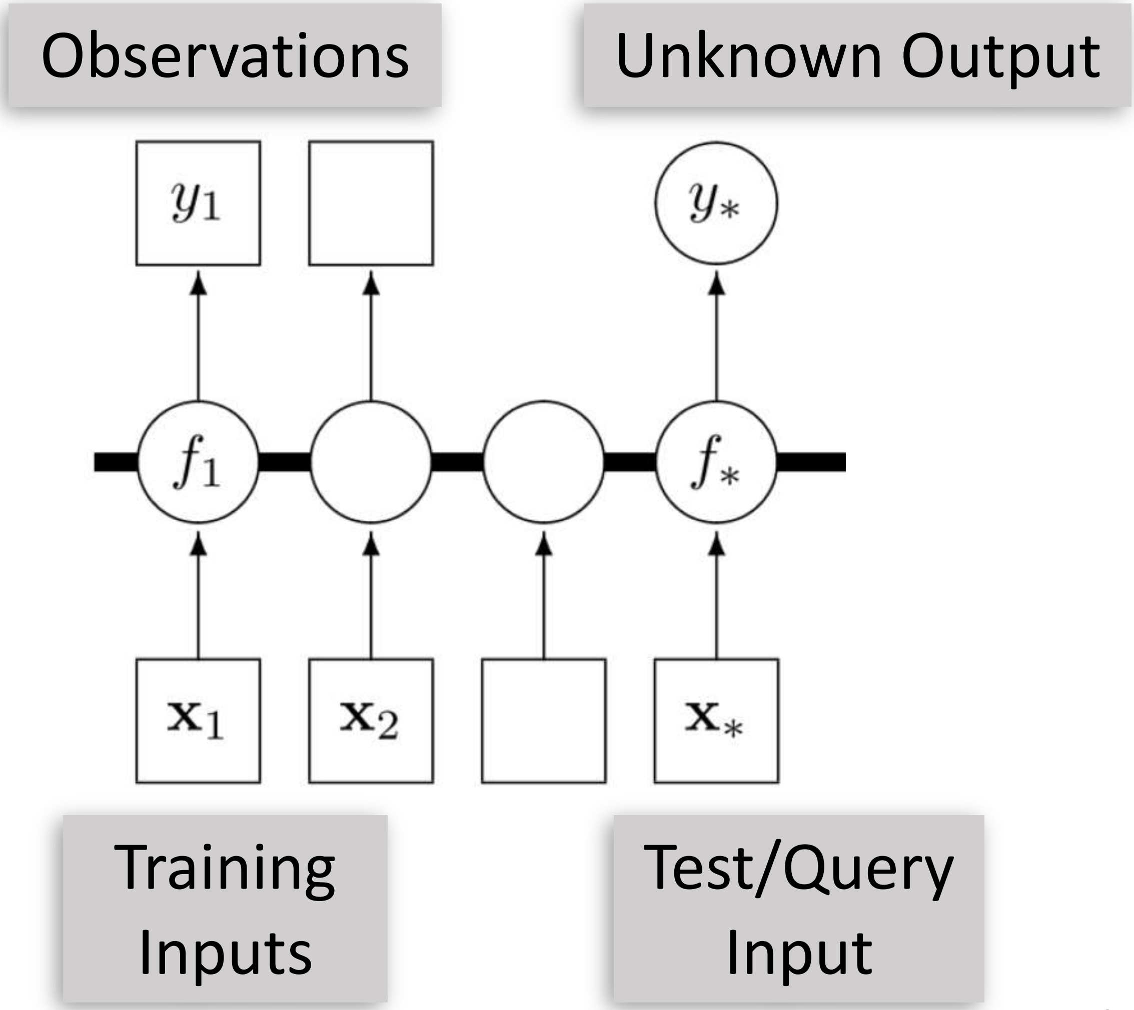


Jakobshavn Glacier
IceBridge 2011
Courtesy of NASA

TASK: Develop a statistical emulator of ice-sheet contributions to sea level rise using using Gaussian Process Modeling

Python/Read

What is Gaussian Process Modeling?



- Non-parametric: there are a distribution of **functions** consistent with observations
- These functions are jointly Gaussian, e.g.

$$f = \mathcal{N}(\mu(x), k(x, x'))$$

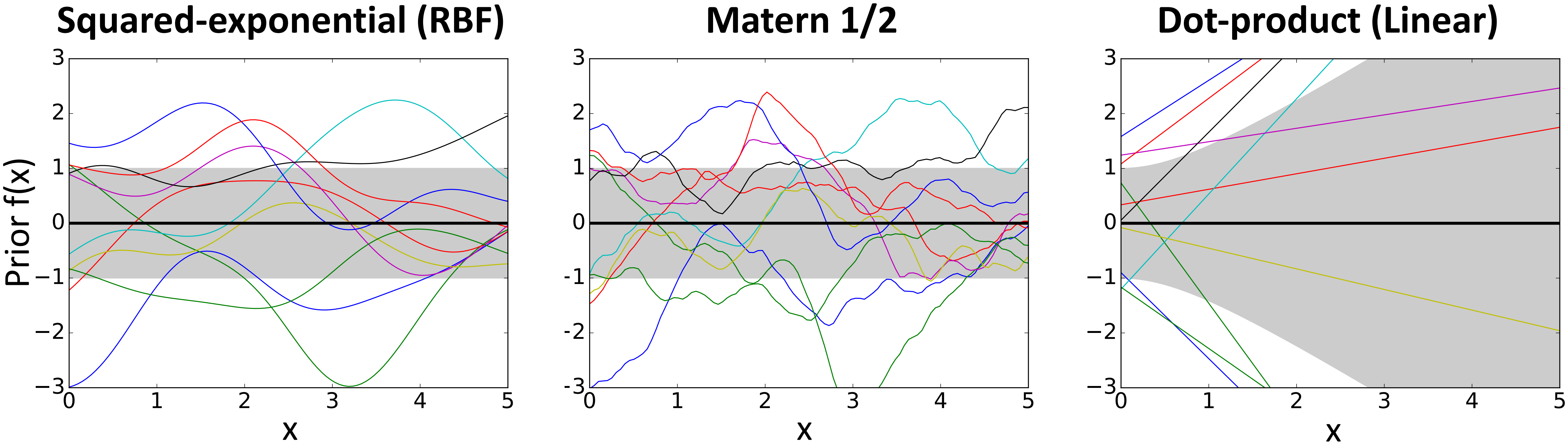
- Uncertainty inherently provided

For more details: Rasmussen and Williams (2006), http://www.gaussianprocess.org/gpml/chapters/RW.pdf

What is Gaussian Process Modeling?

Covariance function general form:

$$k = Amplitude * F\left(\frac{distance(x, x')}{Length\ Scale}\right)$$

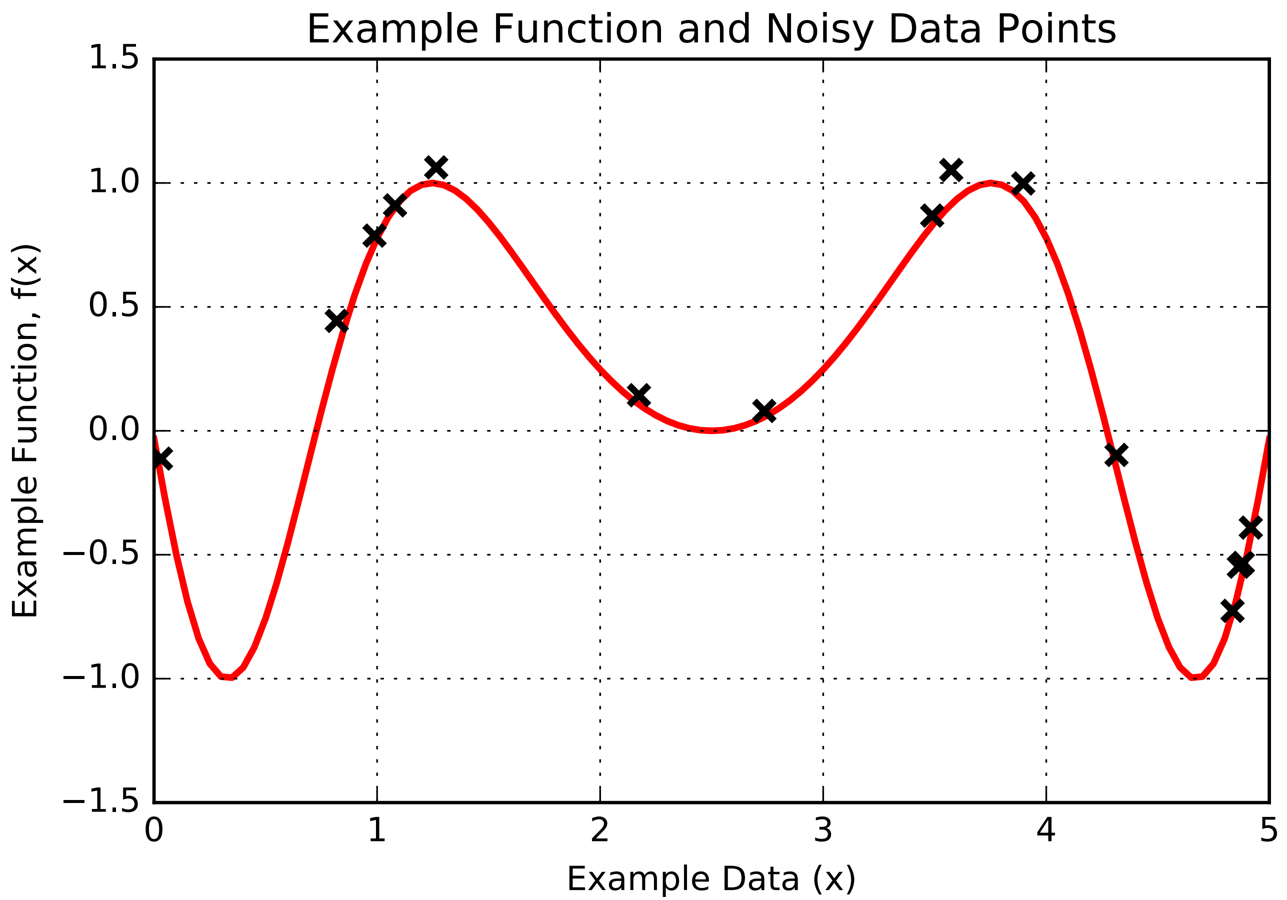


Example: Simple trigonometric curve

— $y = \sin((x - 2.5)^2)$

X training obs. + noise

- Use gpflow package to optimize the hyperparameters of the covariance function
- Try a squared-exponential k



Example: Simple trigonometric curve

$y = \sin((x - 2.5)^2)$

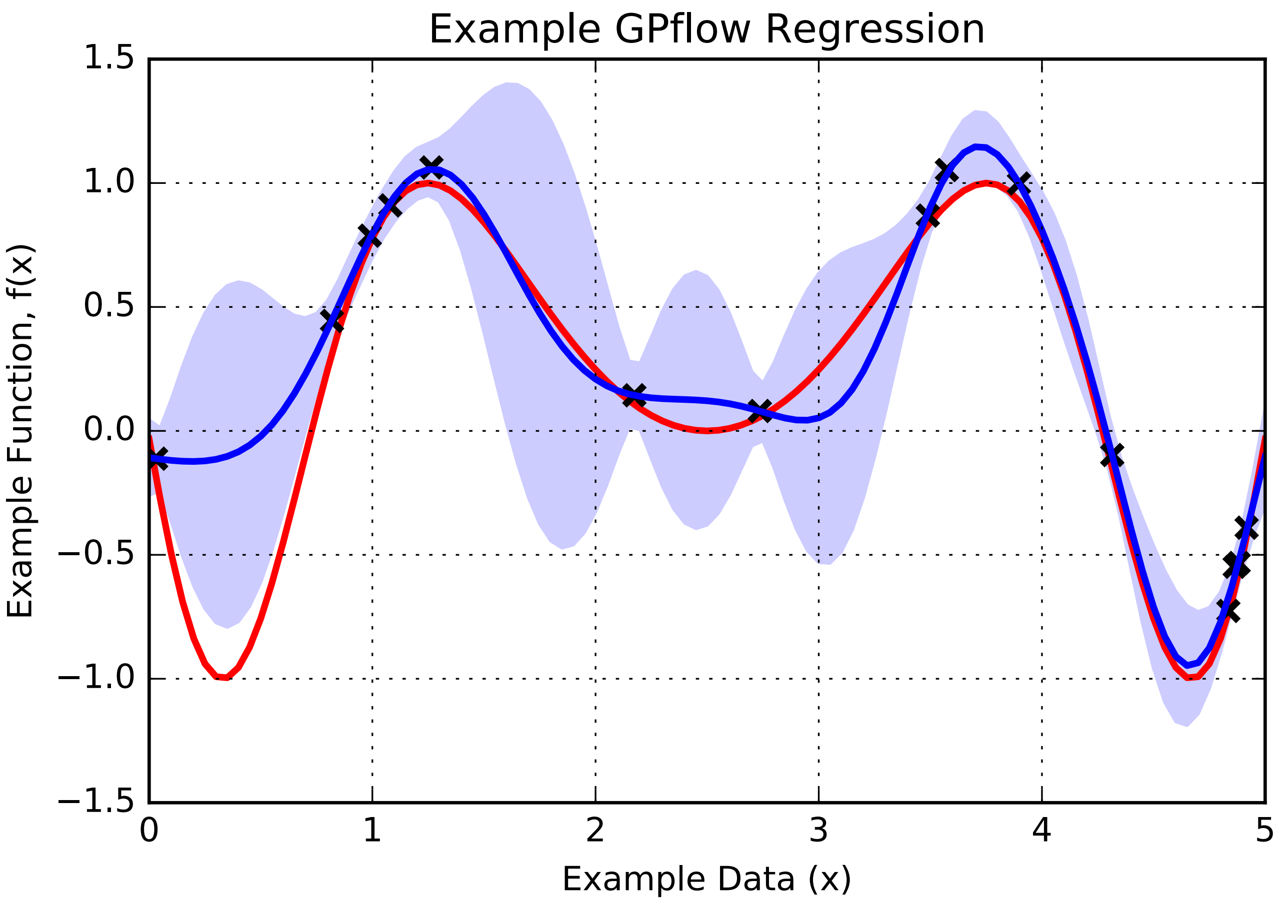
X training obs. + noise

μ of fit model

2σ of fit model

$$Amplitude = 0.38$$

$$Length\ Scale = 0.34$$



Example: Simple trigonometric curve

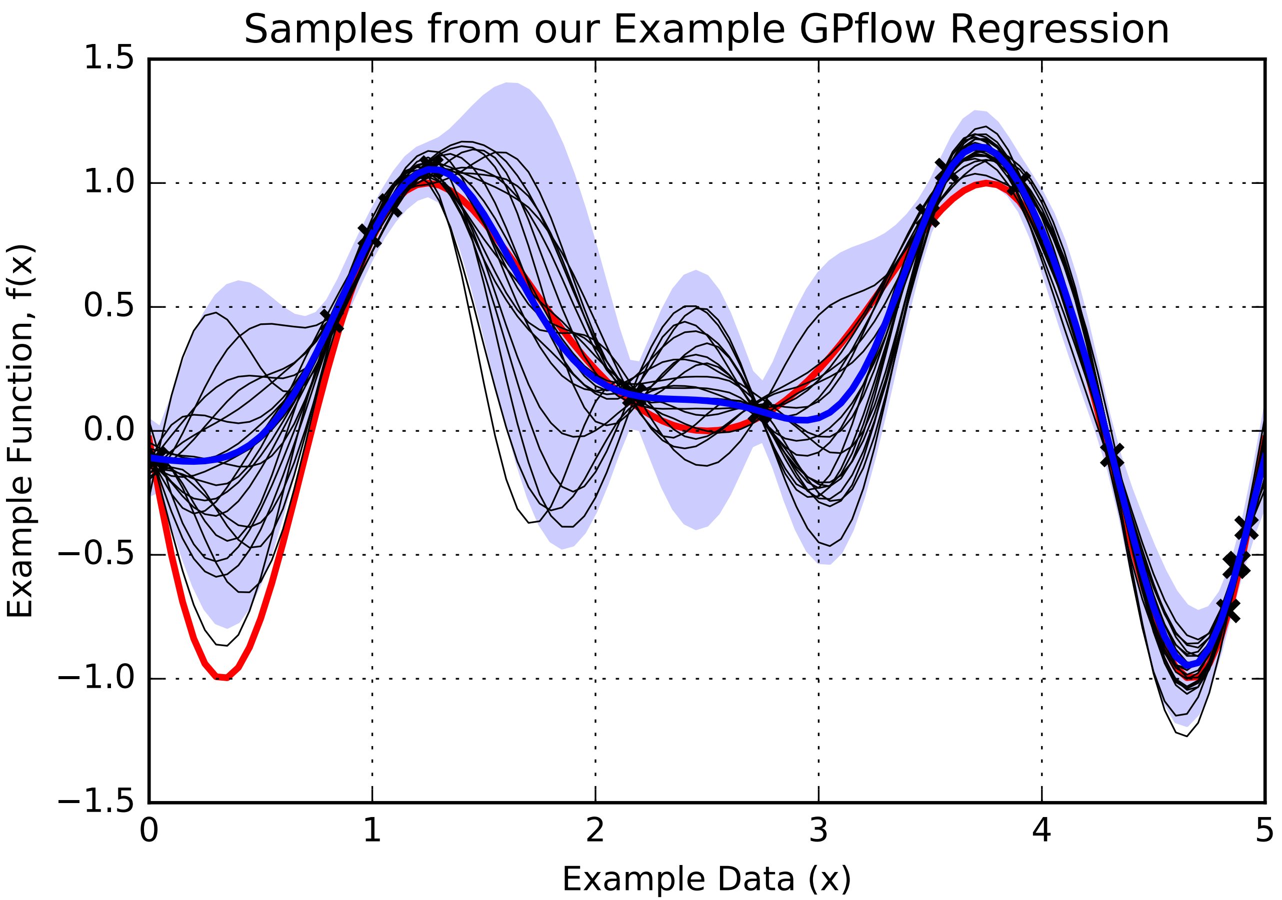
$y = \sin((x-2.5)^2)$

X training obs. + noise

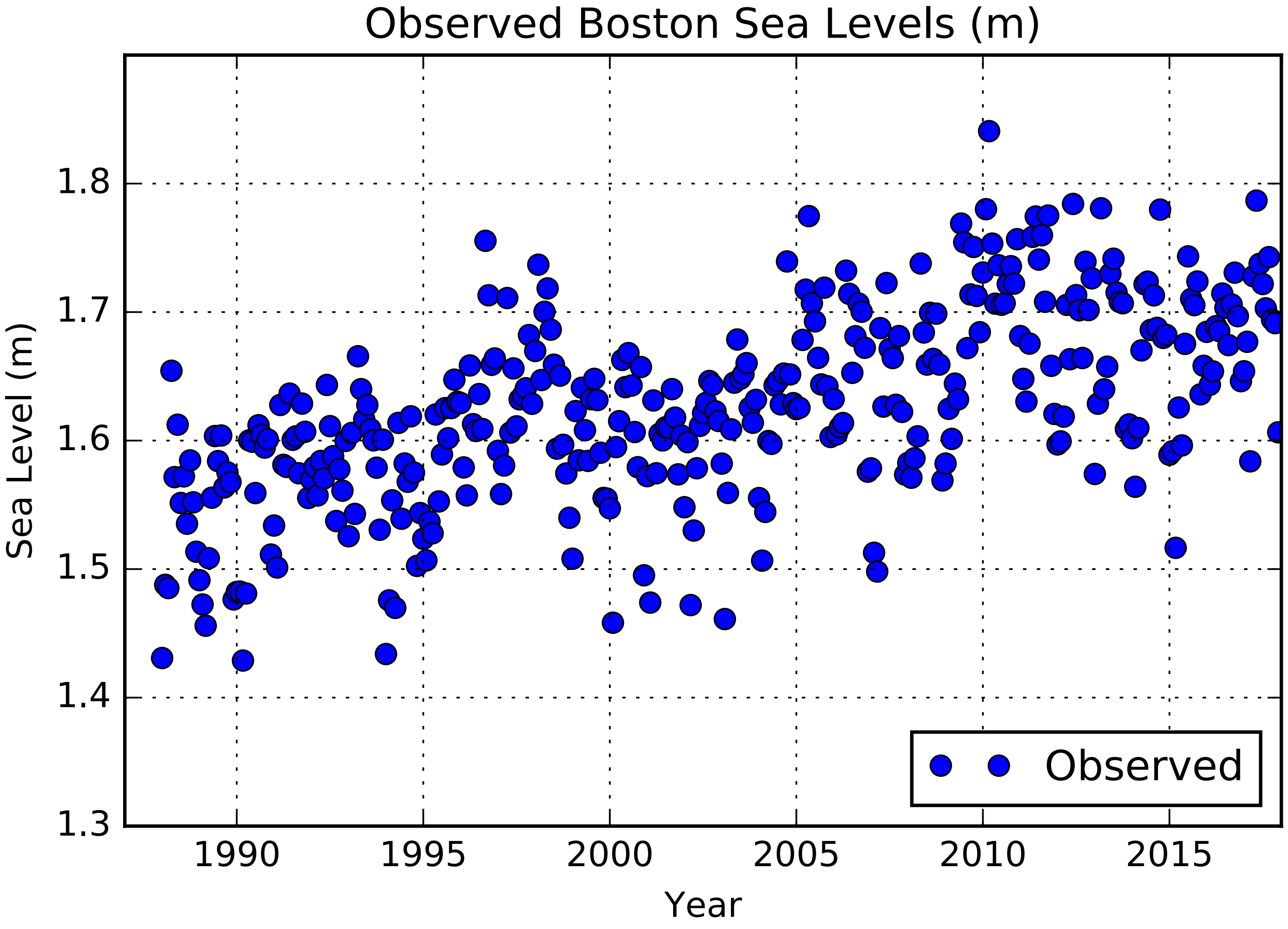
μ of fit model

2*σ* of fit model

samples from model



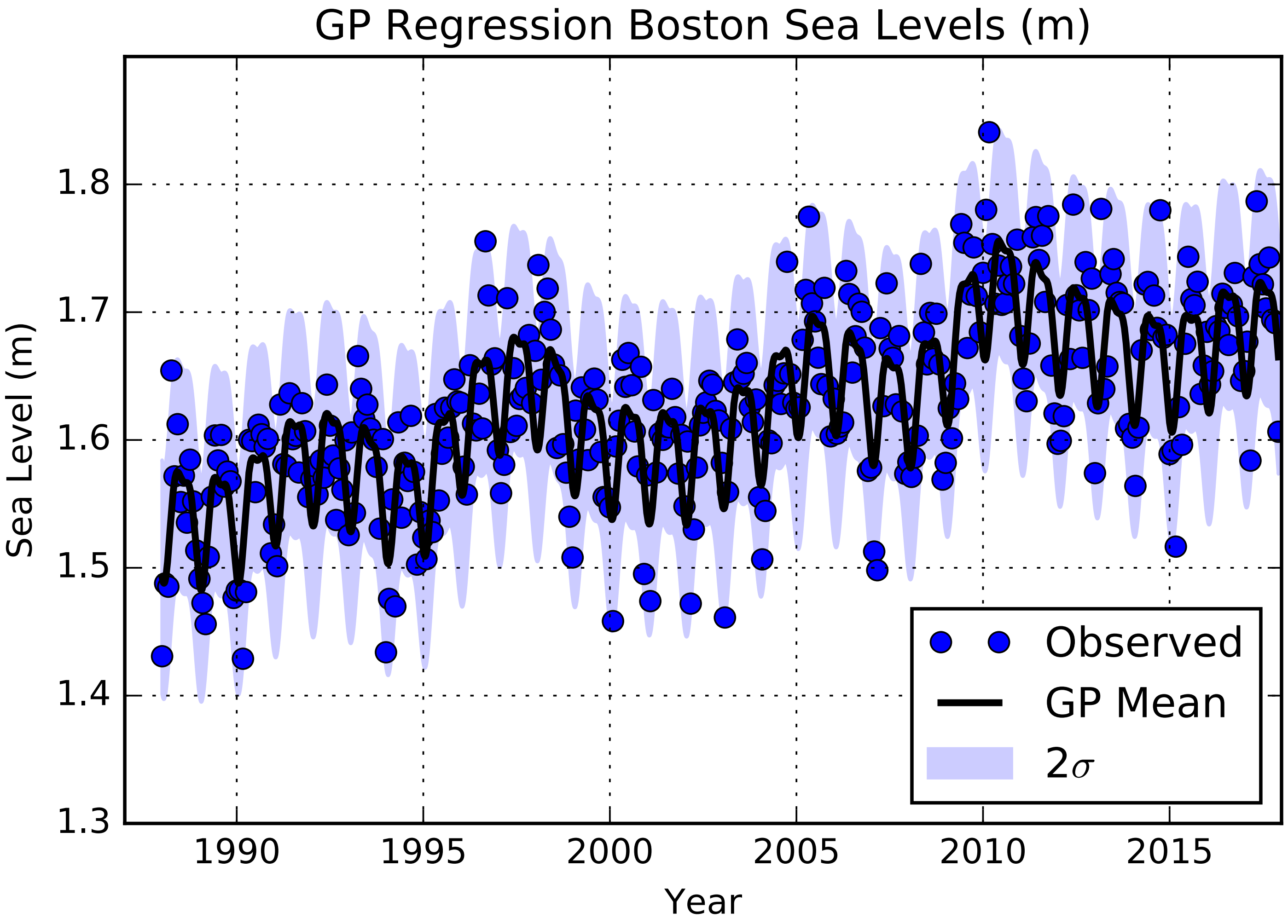
gpflow: GP Regression on Boston Sea Levels



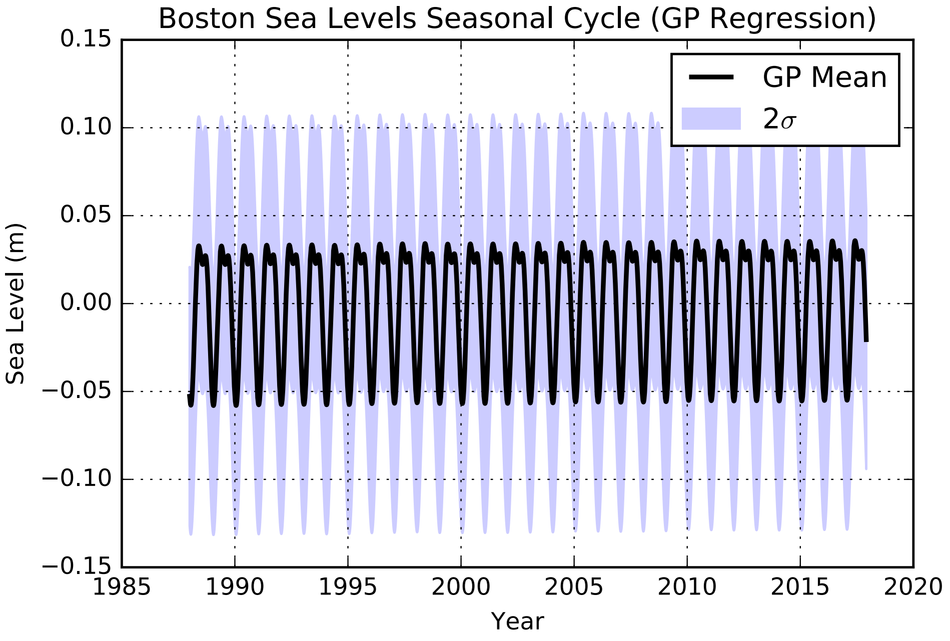
Boston Tide Gauge

Courtesy of NOAA

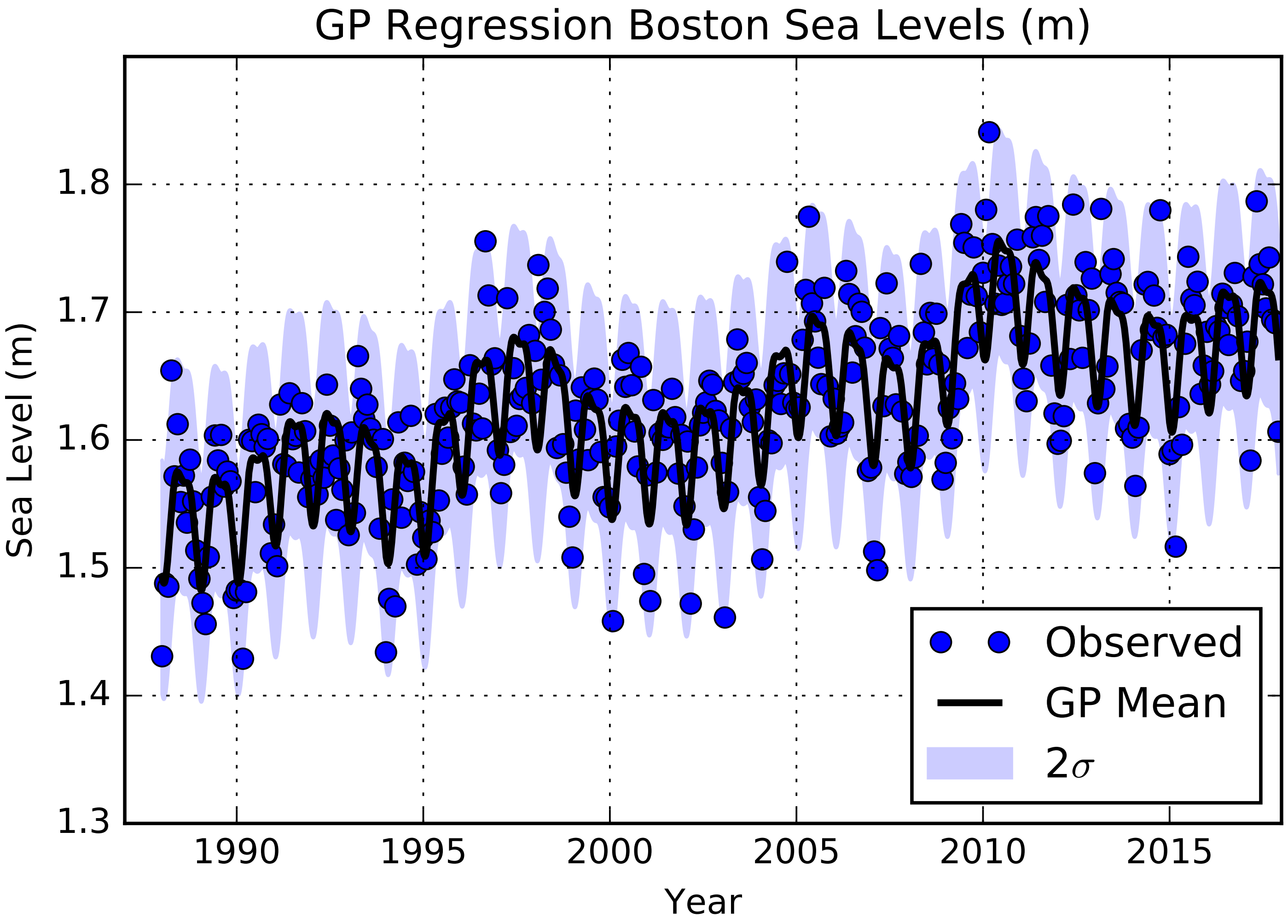
gpflow: GP Regression on Boston Sea Levels



$$k = Linear + Matern + Periodic + Constant$$

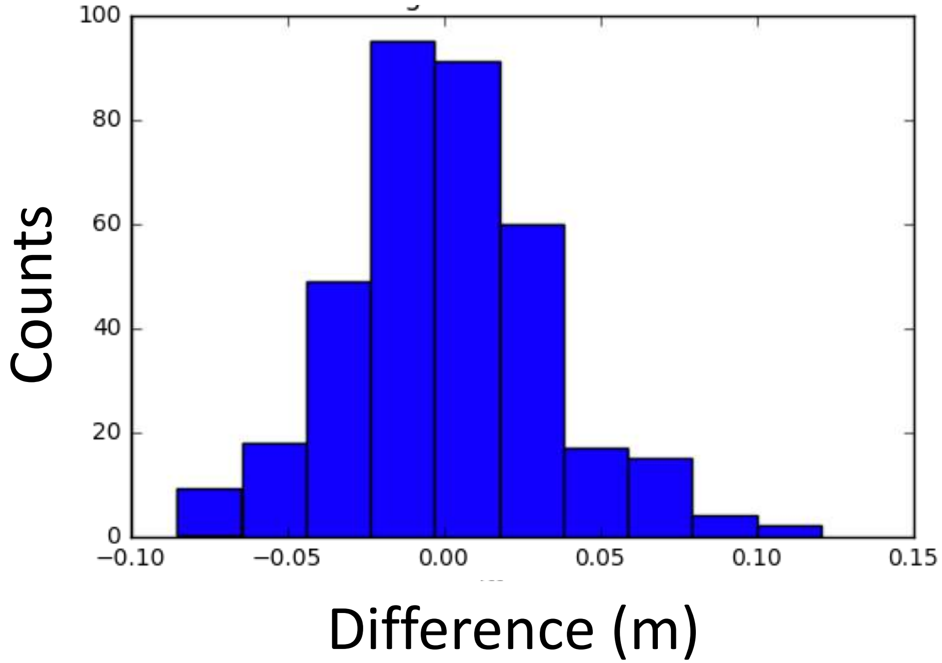


gpflow: GP Regression on Boston Sea Levels



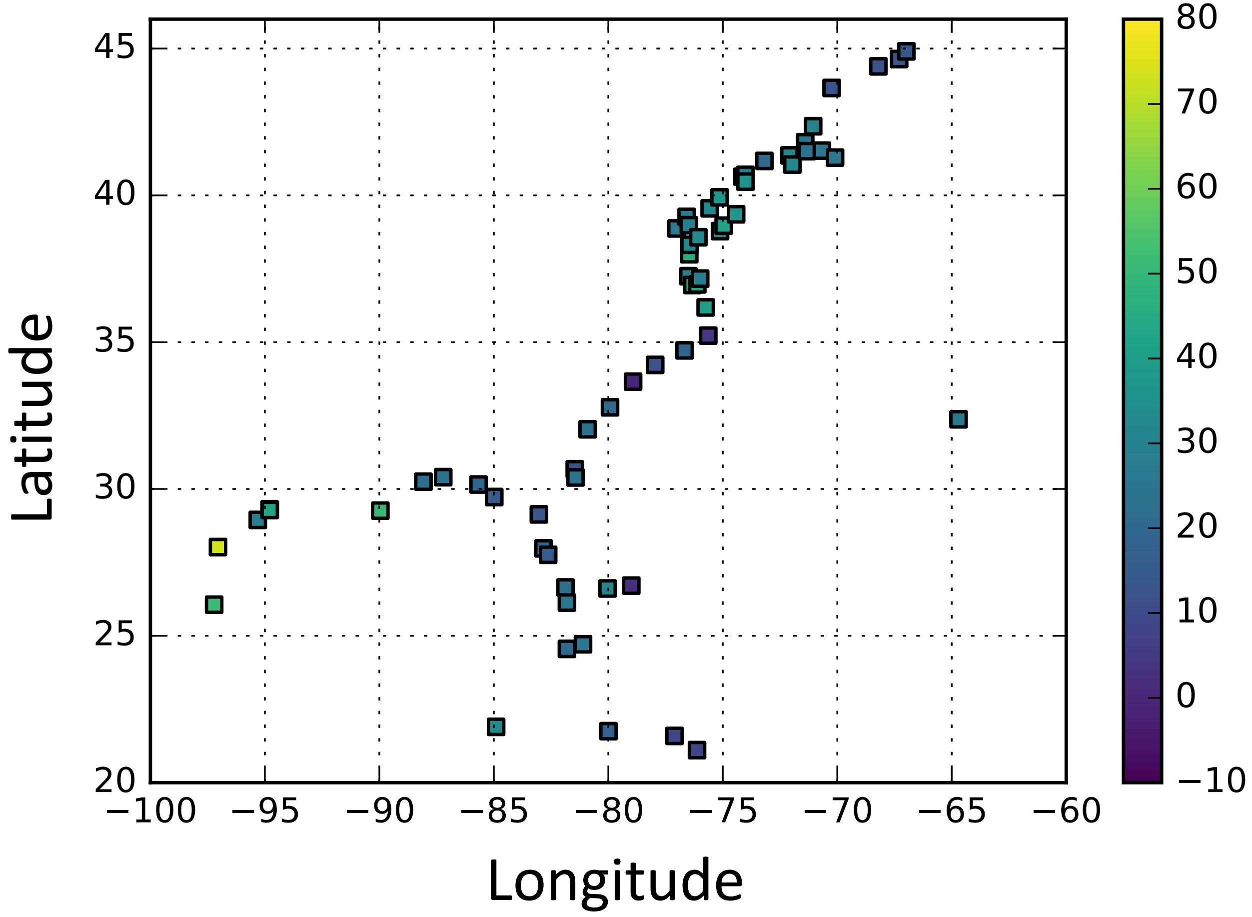
$$k = Linear + Matern$$
$$+ Periodic + Constant$$

Observations – predicted mean



scikit-learn: Spatial Regression on a Sphere

Observed datum offsets at Tide Gauges (mm)

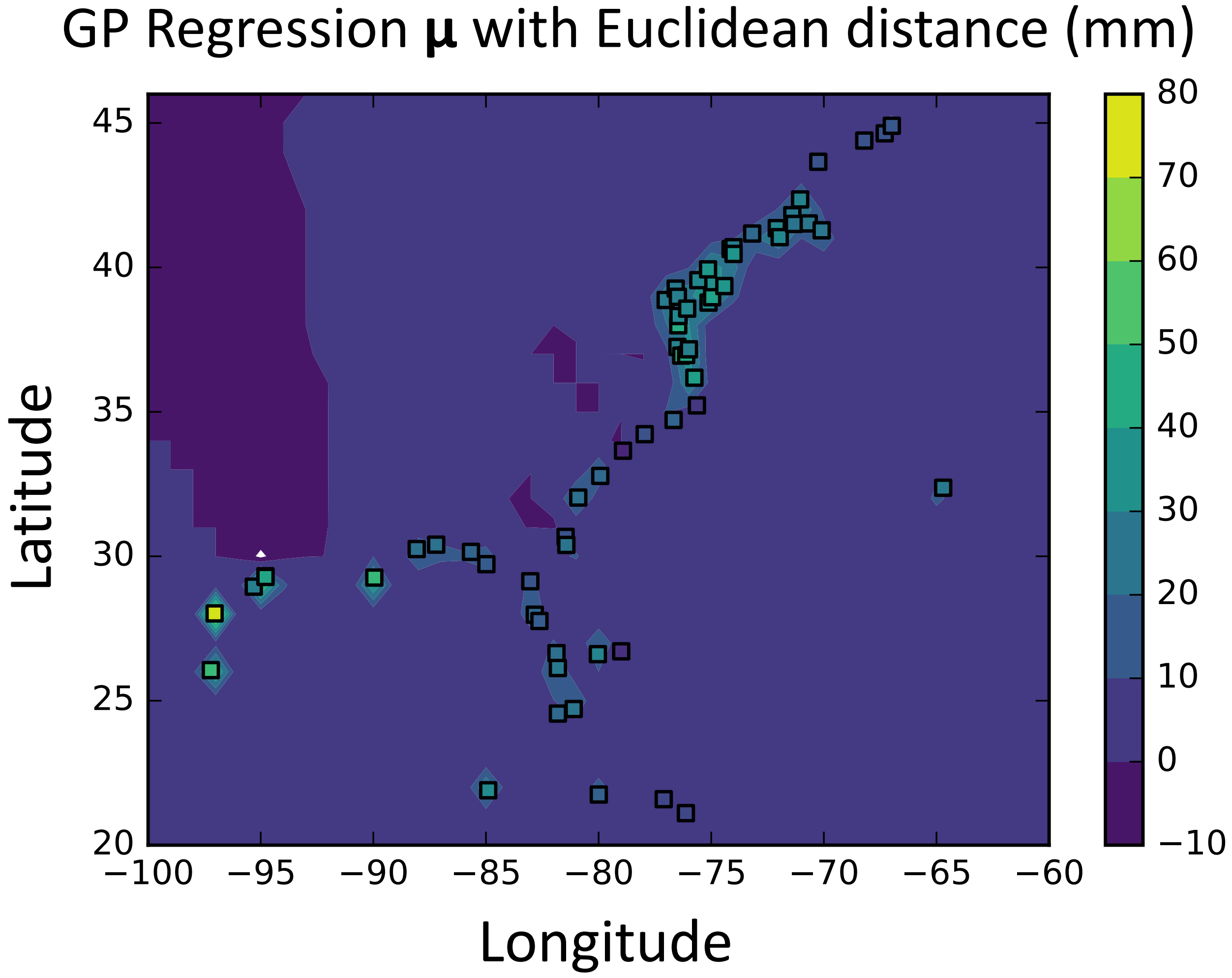


- Goal: Map geolocations (lat/lon) to "datum" differences across the east and gulf coasts

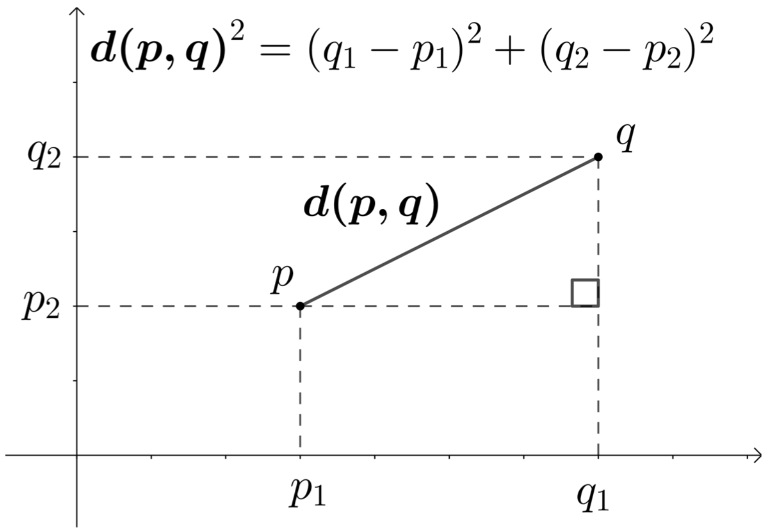
- How do we defined the distance, d(x,x'), in k for this model?

For more on datums, see: https://tidesandcurrents.noaa.gov/datum_options.html

scikit-learn: Spatial Regression on a Sphere

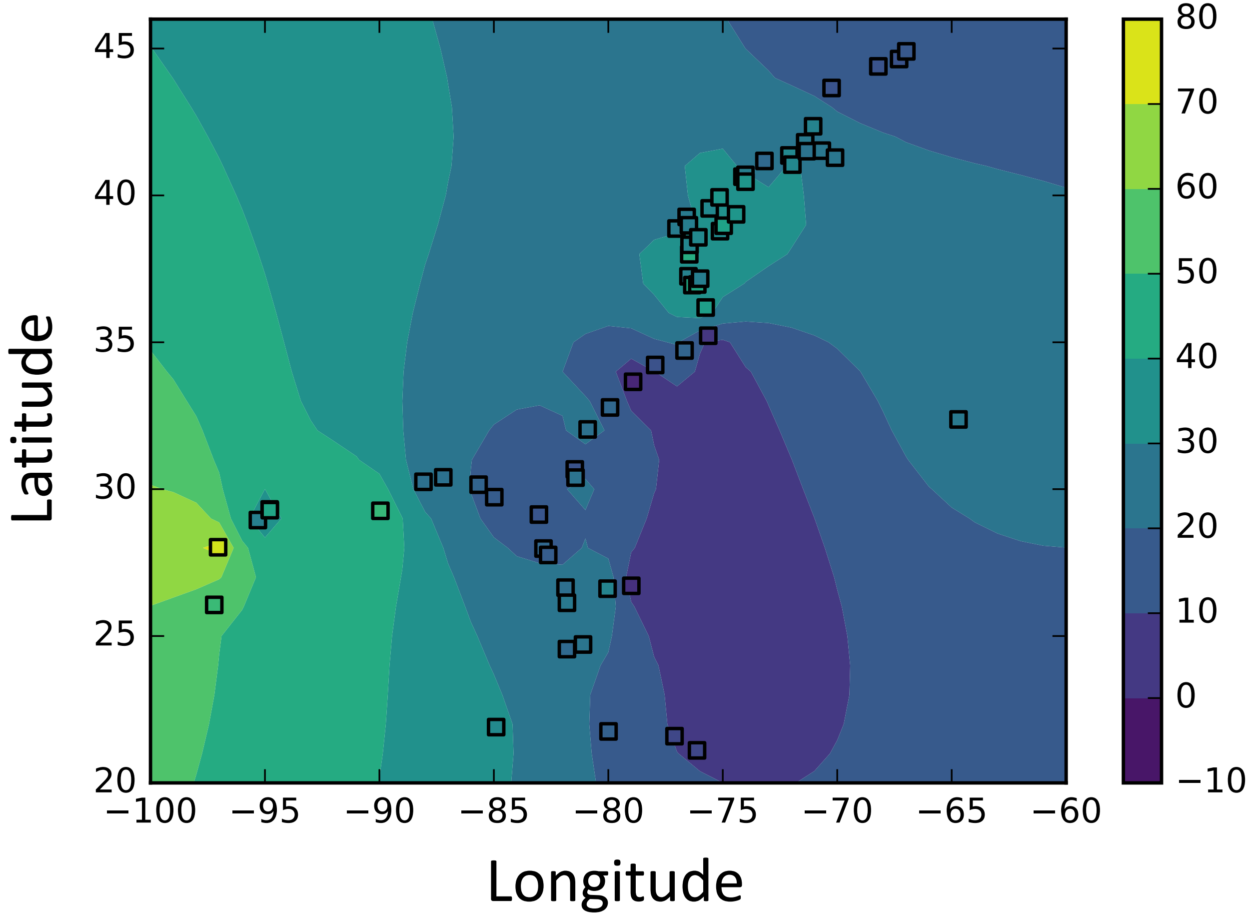


- Goal: Map geolocations (lat/lon) to “datum” differences across the east and gulf coasts

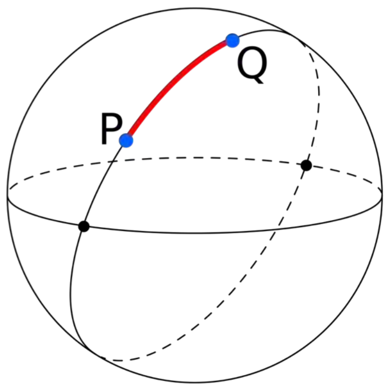


scikit-learn: Spatial Regression on a Sphere

GP Regression **μ** with Haversine distance (mm)

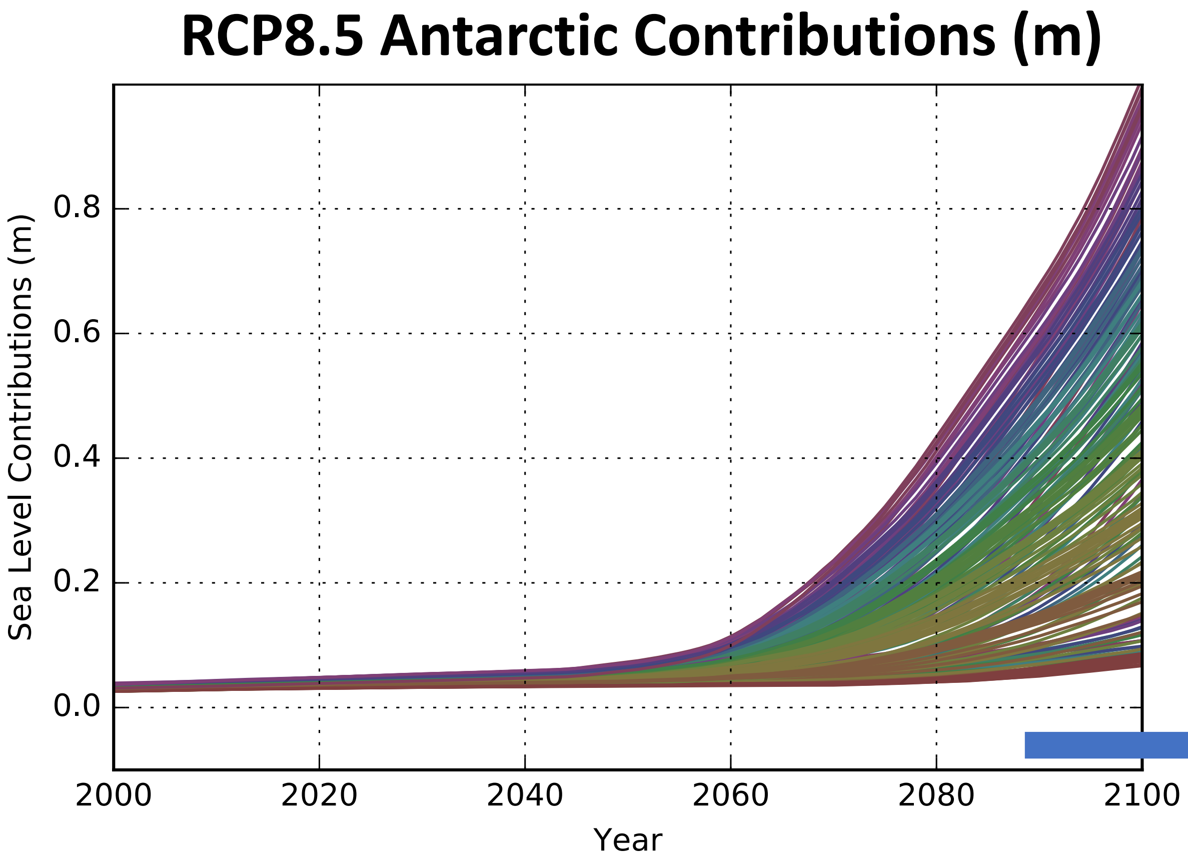


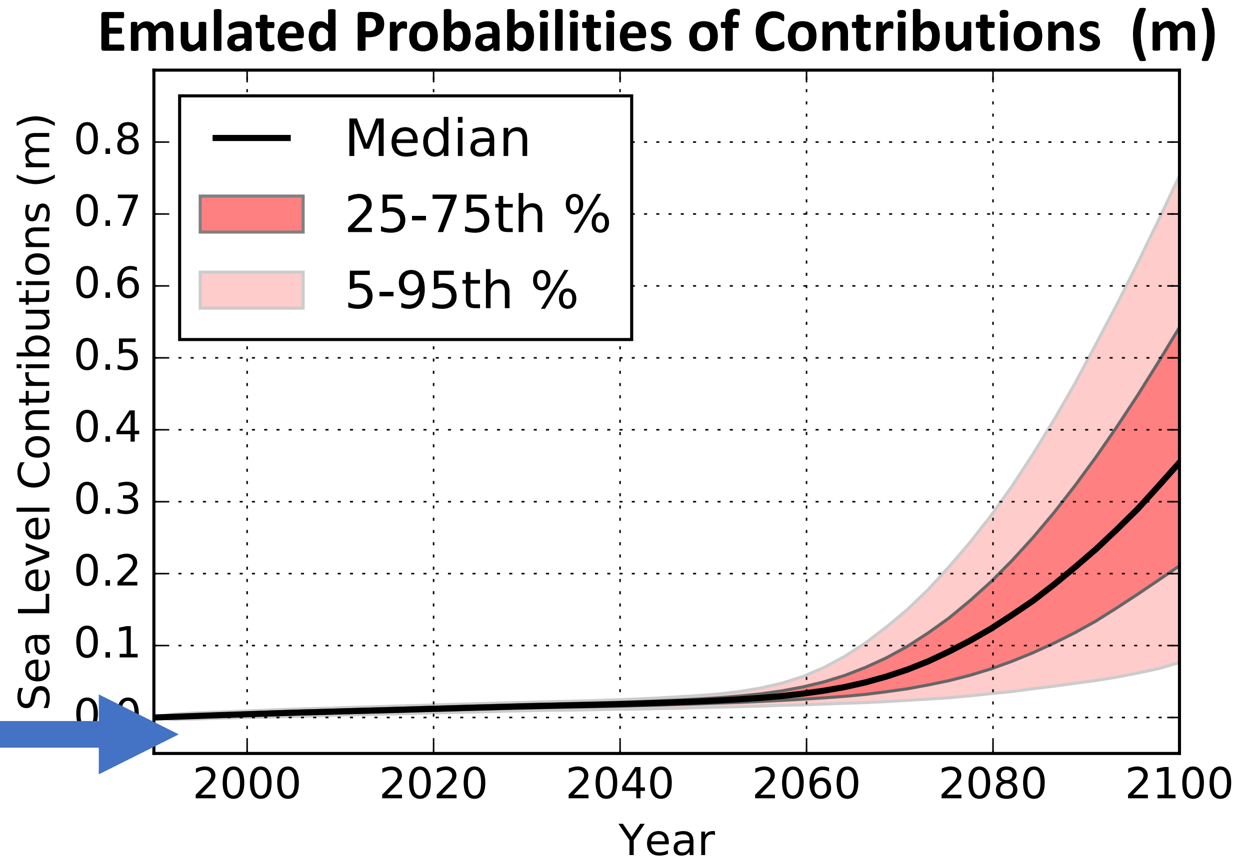
- Goal: Map geolocations (lat/lon) to "datum" differences across the east and gulf coasts



Haversine package:
https://github.com/mapado/haversine

Ice-Sheet Model Emulator





Using gpflow to construct an emulator, we condition on modern and paleoclimate observations to construct a probability distribution of projections

Reflections on Python as a Self-Teaching Tool

1. Python is an effective tool for teaching oneself new concepts. It particularly excels in statistics, machine learning, or other subjects which may be intuitively explored through graphics

Want to learn and rapidly adopt a new tool?
Use Python!

2. Implementing a new package is easy to do, often requiring just a few lines of code to explore a working example

Reflections on Python as a Self-Teaching Tool

3. Many Python packages have flexibility and transparency, so they can be adapted as needed

→ Take the leap… **Leaps facilitate learning!**

4. Don't rely on a single package. Explore as many as you have bandwidth for: push boundaries, discover gaps, develop ideas

→ I often learned most when I realized a package out-of-the-box *wasn't* capable of what I wanted to do

Find this presentation, a simple gpflow tutorial, and more resources at http://danielgilford.com

daniel.gilford@rutgers.edu @danielgilford

RUTGERS
Institute of Earth, Ocean, and Atmospheric Sciences

Support From:
NSF NASA

Additional Resources

Rasmussen and Williams (2006):
http://www.gaussianprocess.org/gpml/chapters/RW.pdf

Gaussian processes for dummies, by Kat Bailey:
http://katbailey.github.io/post/gaussian-processes-for-dummies/

Kernel cookbook, by David Duvenaud:
https://www.cs.toronto.edu/~duvenaud/cookbook/