# Bayesian inference for high-dimensional nonstationary Gaussian processes

**Mark D. Risser**

Climate and Ecosystem Sciences Division

Berkeley National Laboratory, Berkeley, CA, USA

Department of Statistics Seminar

*Clemson University*

25 January 2020

## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

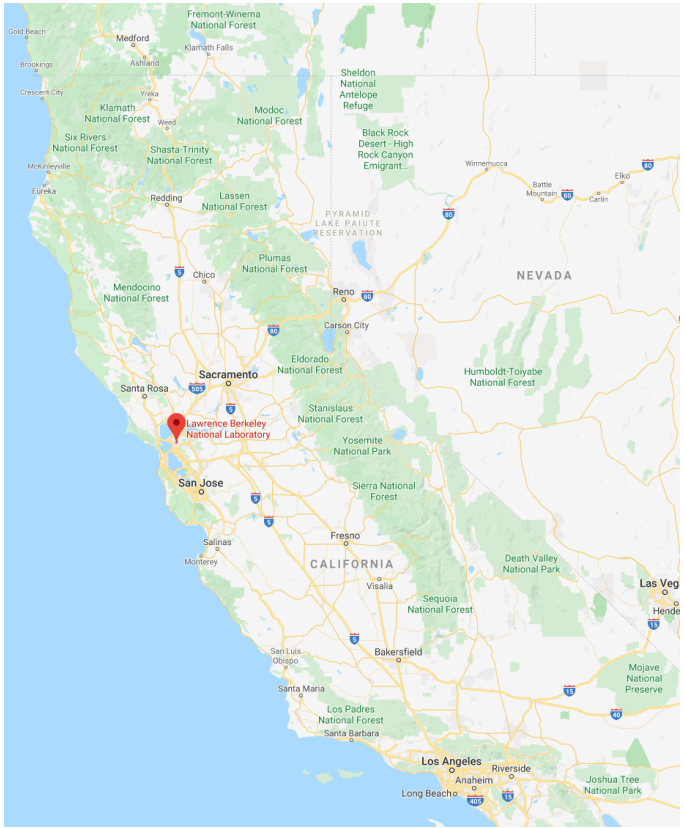
## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

**Mission statement:** "Bringing science solutions to the world"

- LBNL conducts scientific research (primarily) on behalf of the US Department of Energy
- Founded in 1931
- Research areas: genomics, biophysics, engineering, climate science, chemical/materials sciences, physics, nuclear, computer science, applied mathematics
- 13 Nobel laureates
- Discovery of 14 elements on the periodic table

# Brief introduction: Lawrence Berkeley National Laboratory




Location: Berkeley, CA

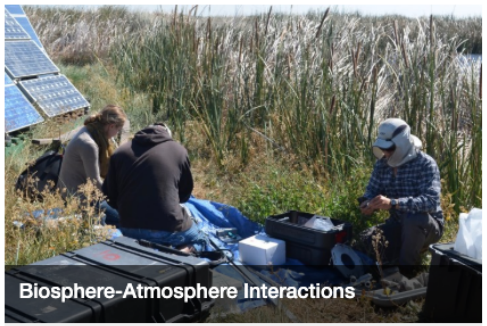
View of Berkeley and SF from LBNL

# Climate and Ecosystem Sciences Division

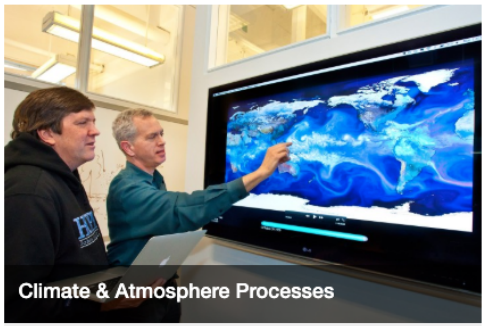
*Mission:* develop the foundational knowledge and capabilities needed to understand, predict, and advance stewardship of Earth's Climate and Ecosystems.

*Vision:* a future where society can make informed decisions about the sustainable use of our planet's resources based on advanced knowledge of the integrated Earth system.

## *Program Domains*




To advance understanding of dynamic biosphere-atmosphere interactions involving greenhouse gases, water and energy, such as respiration, photosynthesis, and ecosystem carbon storage.




Climate and Atmosphere Processes scientists study the processes that drive variability and change in the atmosphere and broader climate system. They develop modeling tools to predict these changes at different time and space scales.


Our mission is to provide decision-relevant insight at the interface of human and natural systems to support resiliency of energy, water, agriculture, and built environments in the face of global and regional change.


Develop a predictive understanding of environmental processes and microbial metabolic diversity that mediate biogeochemical cycles, and develop robust environmental solutions.

## Undergraduate and graduate research opportunities

1. Science Undergraduate Laboratory Internship (SULI): 10-16 week internships (`https://education.lbl.gov/internships/suli/`)
2. Berkeley Lab Undergraduate Research (BLUR): paid; includes graduate students (`https://education.lbl.gov/internships/blur/`)
3. Science Graduate Student Research (SCGSR): spend 3-12 months on thesis-related research with a DOE scientist (`https://education.lbl.gov/internships/scgsr/`)

## Postdoctoral fellowship at LBNL

- Two+ years; join existing projects related to climate, climate extremes, climate change
- After: (a) tenure-track faculty job **or** (b) transition to a LBNL research scientist

## Research as a statistician at a national lab

- Not a consulting position $\rightarrow$ goal is to develop long-term collaborative research
- Data-driven approach to research: development of new methods motivated by science questions
- Exciting opportunity to apply advanced statistical techniques to cutting edge, highly relevant research
- No formal statistics group (at LBNL), but connections with UC Berkeley stat department: seminars, reading groups, teaching opportunities

## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

## What is a Gaussian process (GP)?

1. "Process" $\rightarrow$ a stochastic process: a mathematical model for a collection of random variables indexed over a continuous domain (for example, time or space)
2. "Gaussian" $\rightarrow$ every finite collection of random variables has a multivariate Normal distribution

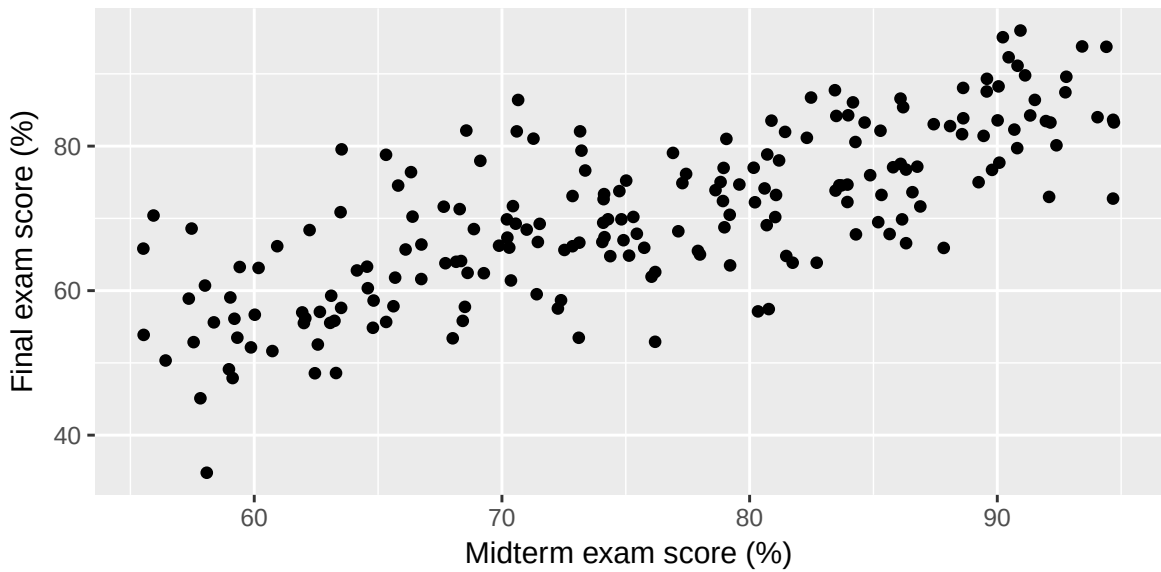
*Notes:*

- Extremely popular tool in statistical modeling
- Spatial and environmental statistics, machine learning, emulation of complex mathematical models
- "Nonparametric," nonlinear regression (as opposed to simple linear regression)

## Detour: simple linear regression (Stat 101)

- Determine the **linear** relationship between a response variable of interest ($y$) and one explanatory variable ($x$)

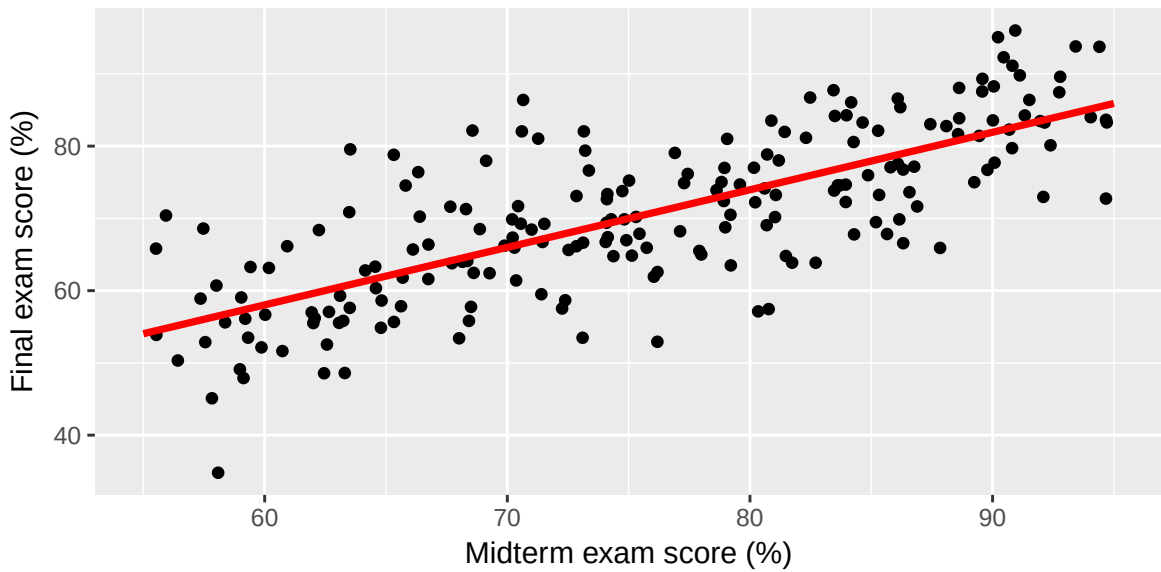
*Midterm exam grades vs. final exam grades*



## Detour: simple linear regression (Stat 101)

- Determine the **linear** relationship between a response variable of interest ($y$) and one explanatory variable ($x$)

*Midterm exam grades vs. final exam grades*

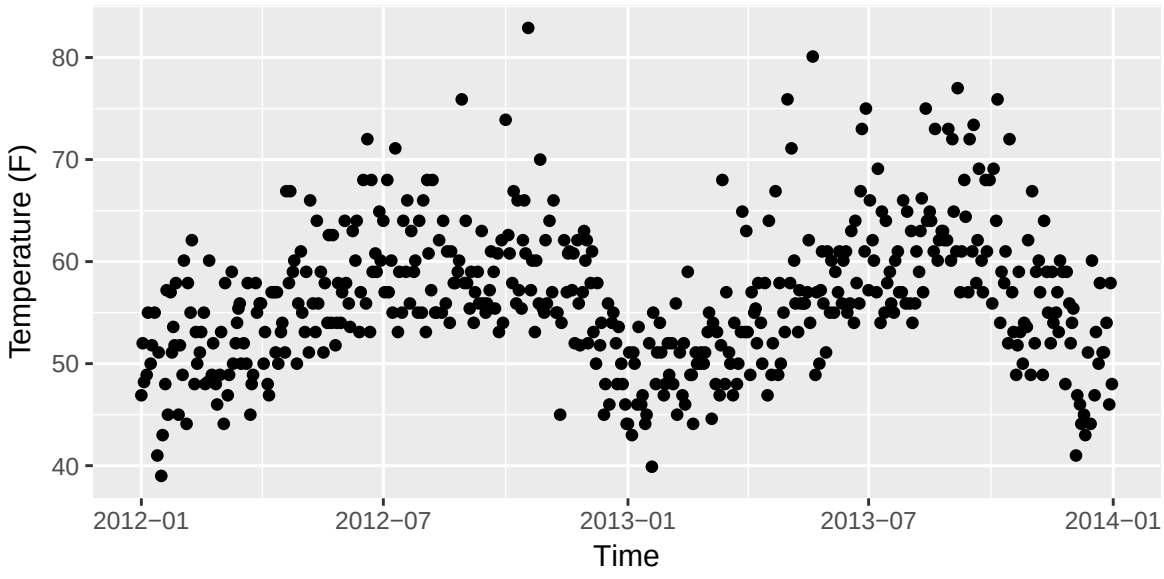



$\rightarrow$ A linear model captures this relationship well

- Statistical parameters: $\beta_0$ (intercept), $\beta_1$ (slope), $\sigma$ (error)

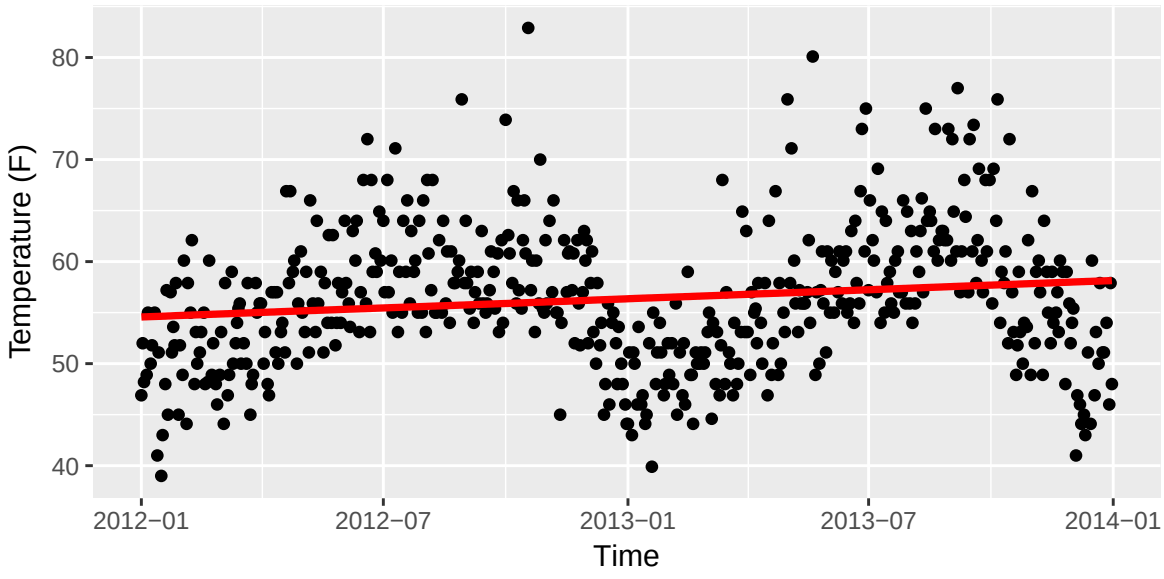
## Detour: simple linear regression (Stat 101)

*Daily temperature at SF airport, 2012-2013*



## Detour: simple linear regression (Stat 101)

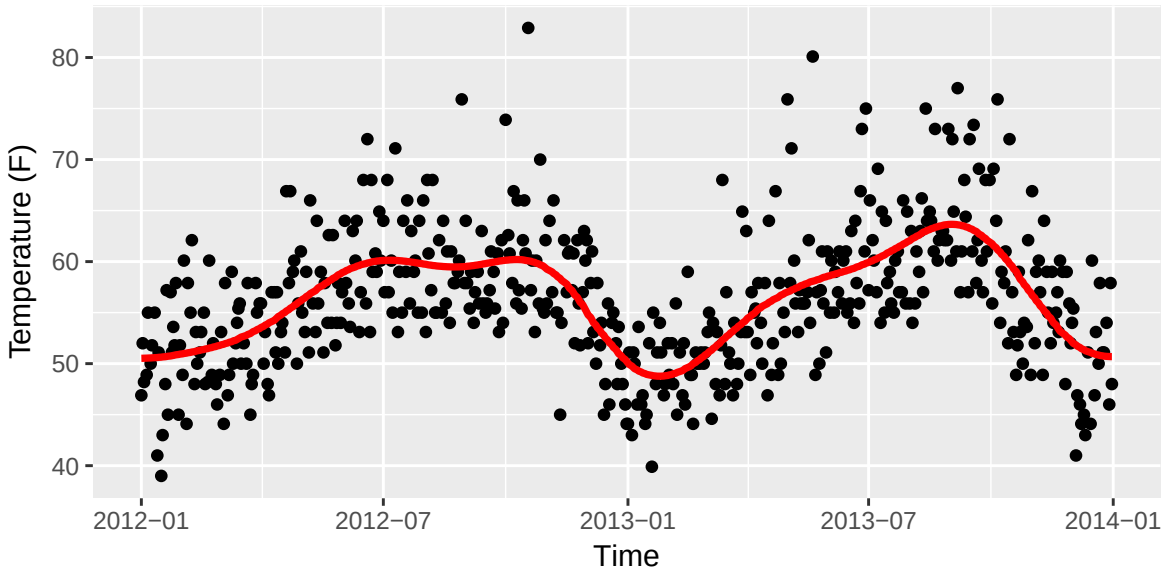
*Daily temperature at SF airport, 2012-2013*




$\rightarrow$ A linear model *does not* capture this relationship well

## Gaussian processes (GPs): or nonlinear regression

*Daily temperature at SF airport, 2012-2013*




- Fit a Gaussian process to these data (red line)
- We have now characterized a *nonlinear* relationship
- Statistical parameters: $\mu$ (mean), $\sigma$ (variance), $\Sigma$ (degree of dependence or smoothness)

## A Bayesian framework for GPs

- Let $z(\cdot)$ be a realization of a stochastic process defined for all $\mathbf{s} \in G \subset \mathbb{R}^d$, with $d \geq 1$ (for a spatial process, $d = 2$)

- Linear mixed model: $z(\mathbf{s}) = y(\mathbf{s}) + \varepsilon(\mathbf{s})$
  - $\rightarrow$ $E[z(\mathbf{s})] = y(\mathbf{s})$; $y(\cdot)$ is a spatially dependent random effect
  - $\rightarrow$ $\varepsilon(\cdot)$ is a spatially independent stochastic error process, distributed as $N(0, \tau^2(\mathbf{s}))$ with $\varepsilon(\cdot) \perp y(\cdot)$
  - $\rightarrow$ Error variance $\tau^2(\cdot)$ known up to parameters $\boldsymbol{\theta}_z$

- Gaussian process model: $y(\cdot) \sim GP\big(\mathbf{x}(\cdot)^\top \boldsymbol{\beta}, C_y(\cdot, \cdot; \boldsymbol{\theta}_y)\big)$
  - $\rightarrow$ $\mathbf{x}(\cdot)^\top \boldsymbol{\beta}$ is a mean function, linear in $\mathbf{x}(\cdot)$
  - $\rightarrow$ $C_y$ is the covariance function:

$$C_y(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}_y) \equiv Cov\big(y(\mathbf{s}), y(\mathbf{s}')\big), \qquad \text{for all } \mathbf{s}, \mathbf{s}' \in G$$

## A Bayesian framework for GPs

- Data model: for $N$ observed locations $\mathcal{S}_O = \{\mathbf{s}_1, ..., \mathbf{s}_N\} \in G$, the random vector $\mathbf{z}_O = [z(\mathbf{s}_1), ..., z(\mathbf{s}_N)]$ has a multivariate Gaussian distribution:

$$p(\mathbf{z}_O|\mathbf{y}_O, \boldsymbol{\theta}_z) = N\big(\mathbf{y}_O, \boldsymbol{\Delta}(\boldsymbol{\theta}_z)\big),$$

  where $\boldsymbol{\Delta}(\boldsymbol{\theta}_z) = diag[\tau^2(\mathbf{s}_1), \ldots, \tau^2(\mathbf{s}_N)]$.

- Latent process model: conditional on parameters, the vector $\mathbf{y}_O$ is distributed as

$$p(\mathbf{y}_O|\boldsymbol{\beta}, \boldsymbol{\theta}_y) = N\big(\mathbf{X}_O\boldsymbol{\beta}, \boldsymbol{\Omega}(\boldsymbol{\theta}_y)\big),$$

  where the elements of $\boldsymbol{\Omega}(\boldsymbol{\theta}_y)$ are $\Omega_{ij} \equiv C_y(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\theta}_y)$

## A Bayesian framework for GPs

- Often useful to integrate over $y(\cdot)$ to arrive at the marginal distribution of the data given parameters:

$$p(\mathbf{z}_O|\boldsymbol{\beta},\boldsymbol{\theta}) = \int p(\mathbf{z}_O|\mathbf{y}_O,\boldsymbol{\theta}_z)p(\mathbf{y}_O|\boldsymbol{\beta},\boldsymbol{\theta}_y)d\mathbf{y}_O$$
$$= N\big(\mathbf{X}_O\boldsymbol{\beta}, \boldsymbol{\Delta}(\boldsymbol{\theta}_z) + \boldsymbol{\Omega}(\boldsymbol{\theta}_y)\big)$$

- The covariance function for the marginalized process is

$$C_z(\mathbf{s},\mathbf{s}';\boldsymbol{\theta}) = C_y(\mathbf{s},\mathbf{s}';\boldsymbol{\theta}_y) + \tau(\mathbf{s})\tau(\mathbf{s}')I_{\{\mathbf{s}=\mathbf{s}'\}}$$

- Last piece is the prior distribution for the parameters: $p(\boldsymbol{\beta},\boldsymbol{\theta})$
- Posterior inference for $\boldsymbol{\beta}$ (mean) and $\boldsymbol{\theta}$ (covariance) based on the marginalized posterior:

$$p(\boldsymbol{\beta},\boldsymbol{\theta}|\mathbf{z}_O) \propto p(\mathbf{z}_O|\boldsymbol{\beta},\boldsymbol{\theta})p(\boldsymbol{\beta},\boldsymbol{\theta})$$

## Posterior prediction

- Let $\mathcal{S}_P = \{\mathbf{s}_1^*, ..., \mathbf{s}_M^*\} \in G$ be a set of locations where we need predictions and $\mathbf{y}_P$ the corresponding process values
- The predictive distribution of interest is

$$p(\mathbf{y}_P|\mathbf{z}_O) = \int_{\boldsymbol{\beta},\boldsymbol{\theta}} p(\mathbf{y}_P|\boldsymbol{\beta},\boldsymbol{\theta},\mathbf{z}_O)p(\boldsymbol{\beta},\boldsymbol{\theta}|\mathbf{z}_O)d\boldsymbol{\beta}d\boldsymbol{\theta}.$$

- $p(\mathbf{y}_P|\boldsymbol{\beta},\boldsymbol{\theta},\mathbf{z}_O)$ available in closed form because of GP assumptions
- Given a set of posterior samples $\{\boldsymbol{\beta}_l, \boldsymbol{\theta}_l : l = 1, \ldots, L\}$ from $p(\boldsymbol{\beta},\boldsymbol{\theta}|\mathbf{z}_O)$ (obtained via MCMC), a Monte Carlo estimate is

$$p(\mathbf{y}_P|\mathbf{z}_O) \approx \sum_{l=1}^{L} p\left(\mathbf{y}_P|\boldsymbol{\beta}_l,\boldsymbol{\theta}_l,\mathbf{z}_O\right)$$

## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

## Nonstationary covariance functions: spatially-varying parameters

*Definition:* A process $y(\cdot)$ is said to be second-order stationary if

$$E[y(\mathbf{s})] = E[y(\mathbf{s}+\mathbf{h})] = \mu,$$

$$\text{Cov}[y(\mathbf{s}), y(\mathbf{s}+\mathbf{h})] = \text{Cov}[y(\mathbf{0}), y(\mathbf{h})] = C(\mathbf{h})$$

where the function $C(\mathbf{h})$, $\mathbf{h} \in \mathbb{R}^d$ is called the covariance function, and specifies the degree of dependence in space.

Implications of a second-order stationary process:

1. The expected (mean) behavior of $y(\cdot)$ is the same, regardless of spatial location $\rightarrow$ not a problem: can include fixed effects in the mean
2. The spatial dependence structure of $y(\cdot)$ is the same, regardless of spatial location $\rightarrow$ an extremely restrictive assumption! Rarely appropriate in practice

## Nonstationary covariance functions: spatially-varying parameters

*Instead*: we want to allow $y(\cdot)$ to be a nonstationary spatial process with covariance function

$$\text{Cov}[y(\mathbf{s}_1), y(\mathbf{s_2})] = C(\mathbf{s}_1, \mathbf{s}_2) \neq C(\mathbf{s}_1 - \mathbf{s}_2)$$

$\rightarrow$ Intuitively: allows the spatial dependence structure to change over space

**Primary limitations of existing nonstationary methods:**

1. Scalability to large data sets
2. Difficult to estimate/computation
3. Limited off-the-shelf software for implementation

## **Convolution-based spatial modeling:** a constructive approach

- Main result: a spatial stochastic process $y(\cdot)$ on $G \subset \mathcal{R}^d$ can be defined by the kernel convolution

$$y(\mathbf{s}) = \int_G K(\mathbf{s} - \mathbf{u}) dW(\mathbf{u})$$

where $W(\cdot)$ is a $d$-dimensional stochastic process and $K(\cdot)$ is a kernel function satisfying $\int_{\mathcal{R}^d} K(\mathbf{u}) d\mathbf{u} < \infty$ and $\int_{\mathcal{R}^d} K^2(\mathbf{u}) d\mathbf{u} < \infty$

- If $W(\cdot)$ is Gaussian, $y(\cdot)$ is a (second-order stationary) Gaussian process with covariance function

$$C(\mathbf{s}, \mathbf{s}') = \int_{\mathcal{R}^d} K(\mathbf{s} - \mathbf{u}) K(\mathbf{s}' - \mathbf{u}) d\mathbf{u}$$

## **Convolution-based spatial modeling:** a constructive approach

- Use a spatially-varying $K_{\mathbf{s}}(\mathbf{u}) = \mathcal{N}_d\big(\mathbf{s}, \boldsymbol{\Sigma}(\mathbf{s})\big)$; then $Y(\cdot)$ is a nonstationary GP with the **closed form** covariance function

$$C(\mathbf{s}, \mathbf{s}') = \left(2\sqrt{\pi}\right)^{-d} \left|\frac{\boldsymbol{\Sigma}(\mathbf{s}) + \boldsymbol{\Sigma}(\mathbf{s}')}{2}\right|^{-1/2} \exp\left\{-Q(\mathbf{s}, \mathbf{s}')\right\}$$

$\rightarrow$ "Distance": $Q(\mathbf{s}, \mathbf{s}') = (\mathbf{s} - \mathbf{s}')^\top \left(\frac{\boldsymbol{\Sigma}(\mathbf{s}) + \boldsymbol{\Sigma}(\mathbf{s}')}{2}\right)^{-1} (\mathbf{s} - \mathbf{s}')$

- A useful generalized nonstationary covariance function is:

$$C^{NS}(\mathbf{s}, \mathbf{s}') = \sigma(\mathbf{s})\sigma(\mathbf{s}') \frac{|\boldsymbol{\Sigma}(\mathbf{s})|^{\frac{1}{4}} \, |\boldsymbol{\Sigma}(\mathbf{s}')|^{\frac{1}{4}}}{\left|\frac{\boldsymbol{\Sigma}(\mathbf{s}) + \boldsymbol{\Sigma}(\mathbf{s}')}{2}\right|^{\frac{1}{2}}} \mathcal{M}_\nu\left(\sqrt{Q(\mathbf{s}, \mathbf{s}')}\right)$$

$\rightarrow \mathcal{M}_\nu(\cdot)$ = Matérn correlation allows smoothness / differentiability of the underlying spatial process to be modeled more flexibly and appropriately

## Convolution-based spatial modeling

Use $C^{NS}$ as the nonstationary covariance function for $y(\cdot)$:

$$C_y(\mathbf{s}, \mathbf{s}'; \boldsymbol{\theta}) = \sigma(\mathbf{s})\sigma(\mathbf{s}') \frac{|\boldsymbol{\Sigma}(\mathbf{s})|^{1/4} |\boldsymbol{\Sigma}(\mathbf{s}')|^{1/4}}{\left| \frac{\boldsymbol{\Sigma}(\mathbf{s}')+\boldsymbol{\Sigma}(\mathbf{s}')}{2} \right|^{1/2}} \mathcal{M}_\nu \left( \sqrt{Q(\mathbf{s}, \mathbf{s}')} \right), \quad \mathbf{s}, \mathbf{s}' \in G$$

- All aspects of covariance vary over space:
  - $\rightarrow$ $\sigma(\cdot)$: spatial standard deviation
  - $\rightarrow$ $\boldsymbol{\Sigma}(\cdot)$: anisotropy (range / direction of dependence)
  - $\rightarrow$ $\tau(\cdot)$: error (nugget) standard deviation (when considering $C_z$)
- *Note:* highly flexible ... but must be regularized somehow!

## General modeling framework for $\tau(\cdot)$ and $\sigma(\cdot)$

Statistical models for strictly positive scalar process: for $\phi \in \{\tau, \sigma\}$,

1. Spatially constant: $\phi(\mathbf{s}) \equiv \phi$ for all $\mathbf{s} \in G$
2. Log-linear regression: use observed spatial covariates to describe variability
3. Low-rank approximation to a stationary GP:

$$\log \boldsymbol{\phi} = \mu_\phi \mathbf{1}_N + \sigma_\phi \mathbf{P}_\phi \mathbf{V}_\phi^{-1/2} \mathbf{w}_\phi$$

   $\rightarrow$ $\mathbf{w}_\phi$: latent process defined on $K$ knot locations ($K << N$)
   $\rightarrow$ $\mathbf{P}_\phi \mathbf{V}_\phi^{-1/2}$: radial basis functions calculated using Matérn cross-correlations (obs vs. knot locations) and correlations (knot locations)
   $\rightarrow$ Knot locations = obs locations implies an exact GP

## General modeling framework for $\boldsymbol{\Sigma}(\cdot)$

Statistical models for positive definite matrix process:

1. Spatially constant: $\boldsymbol{\Sigma}(\mathbf{s}) \equiv \boldsymbol{\Sigma}$ for all $\mathbf{s} \in G$
2. Covariance regression: $\boldsymbol{\Sigma}(\mathbf{s}) = \boldsymbol{\Psi} + \boldsymbol{\Gamma}\mathbf{x}(\mathbf{s})[\boldsymbol{\Gamma}\mathbf{x}(\mathbf{s})]^\top$
3. Componentwise regression: decompose $\boldsymbol{\Sigma}(\mathbf{s}) = \boldsymbol{\Gamma}(\mathbf{s})\boldsymbol{\Lambda}(\mathbf{s})\boldsymbol{\Gamma}(\mathbf{s})^\top$,

$$\boldsymbol{\Lambda}(\mathbf{s}) = \begin{bmatrix} \lambda_1(\mathbf{s}) & 0 \\ 0 & \lambda_2(\mathbf{s}) \end{bmatrix}, \quad \boldsymbol{\Gamma}(\mathbf{s}) = \begin{bmatrix} \cos\gamma(\mathbf{s}) & -\sin\gamma(\mathbf{s}) \\ \sin\gamma(\mathbf{s}) & \cos\gamma(\mathbf{s}) \end{bmatrix}$$

   and assign GLMs to each of $\lambda_1(\mathbf{s})$, $\lambda_2(\mathbf{s})$, and $\gamma(\mathbf{s})$
4. Nonparametric regression: apply the low-rank approximation to a stationary GP to transformed $\lambda_1(\mathbf{s})$, $\lambda_2(\mathbf{s})$, and $\gamma(\mathbf{s})$

## Novel framework for modeling nonstationarity

Using closed form covariance function: specify an appropriate statistical model for $\tau(\cdot)$, $\sigma(\cdot)$, $\mathbf{\Sigma}(\cdot)$

1. Highly flexible: spatial variability in covariance parameters can specified deterministically (covariates / basis functions) or stochastically
2. Parsimonious representation of nonstationary covariance function (particularly using regression-based approaches) $\rightarrow$ aids computation / MCMC
3. Interpretable summaries of *why* spatial process of interest is nonstationary (using covariates)

## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

## Limitations of applying GPs to modern data sets

- Calculations involving the multivariate Normal distribution for $N$ spatial locations $\rightarrow$ require $\mathcal{O}(N^2)$ memory and $\mathcal{O}(N^3)$ time complexity
- Particularly problematic for nonstationary GPs: high-dimensional parameter spaces
- Prediction: computational complexity even worse for large $M$
- Diverse literature on approximate GP methods (see Heaton et al., 2018, for a review) $\rightarrow$ none use nonstationary covariance functions

## Vecchia approximation (Vecchia, 1988)

Computational shortcuts via conditioning on subsets of the data:

- Re-write the multivariate Normal distribution as a product of conditional distributions:

$$p(\mathbf{y}_O) = p(y_1) \prod_{i=2}^{N} p\left(y_i | \mathbf{y}_{h(i)}\right)$$

  where $h(i) = (1, \ldots, i-1) \rightarrow$ but, no computational savings yet

- Vecchia (1988): replace $h(i)$ with subvectors $g(i) \subset h(i)$, where $g(i)$ usually refers to the indices that are "near" $\mathbf{s}_i$

$$\widehat{p}(\mathbf{y}_O) = p(y_1) \prod_{i=2}^{N} p\left(y_i | \mathbf{y}_{g(i)}\right)$$

- $\widehat{p}(\mathbf{y}_O)$ converges to $p(\mathbf{y}_O)$ as the $g(i)$ approach $h(i) \rightarrow$ prefer small conditioning vectors for computational efficiency ($|g(i)| \approx 20$ or 30)

## Vecchia approximation (Vecchia, 1988)

Two specific methods: nearest neighbor GP for the response (NNGP-R; Datta et al., 2016) and sparse general Vecchia (SGV; Katzfuss et al., 2018)

- Extend the Vecchia framework to joint distribution of $(\mathbf{y}_O, \mathbf{z}_O)$:

$$\widehat{p}(\mathbf{y}_O, \mathbf{z}_O) = \prod_{i=1}^{N} \left[ p\left(y_i | \mathbf{y}_{q_y(i)}, \mathbf{z}_{q_z(i)}\right) \times p(z_i | y_i) \right]$$

  $\rightarrow$ Assumes $z_i$ is conditionally independent of $\mathbf{z}_{-i}$ given $y_i$

  $\rightarrow$ For each $j \in g(i)$: determine whether condition on $z_j$ or $y_j$

- NNGP-R and SGV specify different algorithms for determining the conditioning set

## Comparison of NNGP-R vs. SGV

- Approximate likelihood for each can be computed in $\mathcal{O}(N)$ time (linear scaling)
- Approximation accuracy is better for SGV relative to NNGP-R
- Katzfuss et al. (2018) show empirically that likelihood calculations for NNGP-R are faster
- SGV performs better in the low signal-to-noise situation
- Prediction: SGV can characterize joint predictions, while NNGP-R can only yield univariate / marginal predictions

## Which to use?

$\rightarrow$ Important tradeoff: accuracy versus speed

$\rightarrow$ Are joint predictions needed?

## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

## The `BayesNSGP` package for R

- Software package for R
- `BayesNSGP` = Bayesian Nonstationary Gaussian Process
- Enables off-the-shelf, Bayesian analysis of spatial data sets
- Implements the methodology discussed so far: flexible specification of spatially-varying covariance parameters with approximate GP methods (NNGP-R and SGV)
- Utilizes functionality from the `nimble` package to make MCMC much easier
- Currently available on CRAN:

```
R> install.packages("BayesNSGP")
```

Two outward-facing user interface functions:

1. `nsgpModel`: set up and run MCMC for a generic nonstationary Gaussian process model

```
nsgpModel( coords, data, constants = list(),
  likelihood  = "fullGP",
  tau_model = "constant", sigma_model = "constant",
  Sigma_model = "constant", mu_model = "constant" )
```

   → Required inputs: spatial coordinates, data, constants (fixed hyperparameters, design matrices, etc.)
   → Specify a likelihood: `"fullGP"` (exact), `"SGV"`, or `"NNGP"`
   → Specify sub-models: `"logLinReg"`, `"approxGP"`, `"covReg"`

2. `nsgpPredict`: corresponding posterior prediction

```
nsgpPredict( model, samples, coords.predict, constants )
```

## General workflow for a specific nonstationary Gaussian process

1. Create a `nimble` "model" object

```
R> model <- nsgpModel( ... )
```

2. Create a MCMC algorithm to fit the model

```
R> mcmc <- configureMCMC(model)
R> mcmc$addSampler(...)
```

   → Customize the MCMC samplers: adaptive Metropolis-Hastings, multivariate/block random walk, slice sampler, etc.

3. Build and compile the model/MCMC

```
R> Rmcmc <- buildMCMC(mcmc) # Build the MCMC
R> Cmodel <- compileNimble(model) # Compile the model in C++
R> Cmcmc <- compileNimble(mcmc, project = model) # Compile in C++
```

4. Run the MCMC

```
R> samples <- runMCMC(Cmcmc, niter = 10000)
```

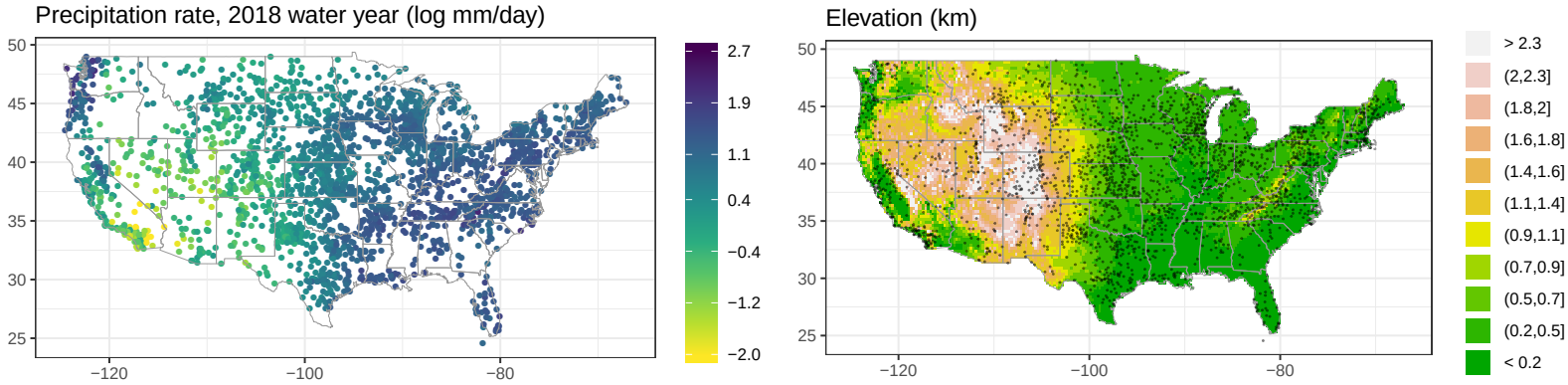
## The BayesNSGP package for R

- I have a few vignette demonstrations (but not on the web yet) – happy to share over email
- Also: see paper in JSCS (open access), https://doi.org/10.1080/00949655.2020.1792472
- Or see Arxiv – https://arxiv.org/abs/1910.14101

## Outline

0. *(Brief introduction: Lawrence Berkeley National Laboratory)*
1. Gaussian process modeling
2. Nonstationary covariance functions: spatially-varying parameters
3. Approximate GP methods
4. The `BayesNSGP` package for R
5. Application

# Application: precipitation rate over CONUS




- In situ measurements of daily precipitation rate (pr) from GHCN database over the contiguous United States (CONUS) for the 2018 water year (October-September)
- $N = 2311$ stations have no missing values over this time period
- CONUS is a highly heterogeneous spatial domain:
  - → Variable topography
  - → Diverse set of phenomena that produce precipitation (atmospheric rivers, tropical cyclones, mesoscale convective systems, etc.)
- Important to fit a nonstationary spatial model to these data!

## Nonstationary spatial model

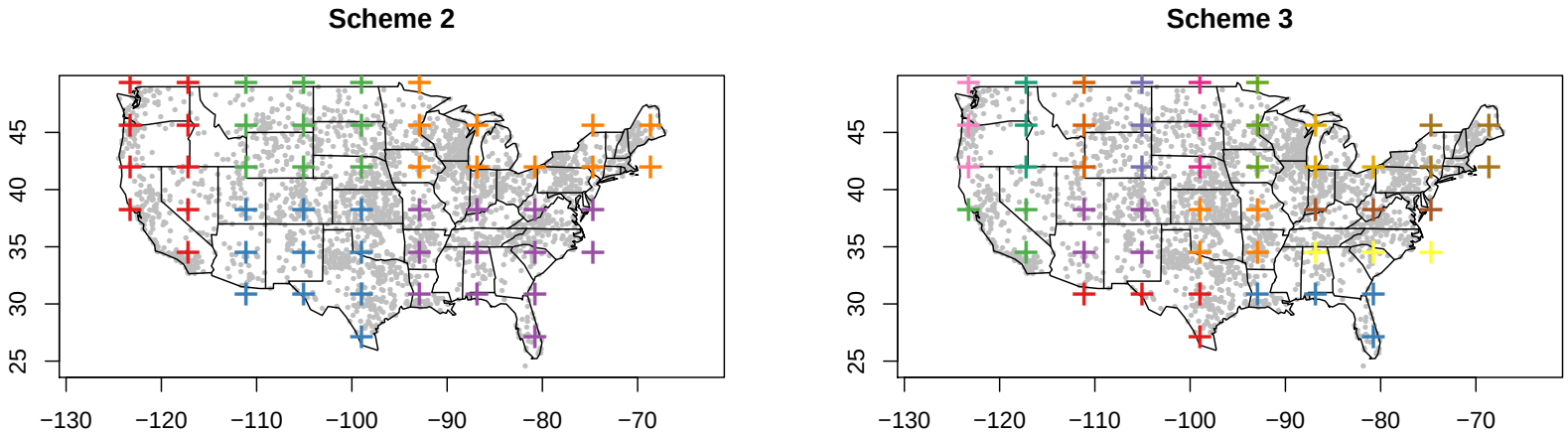
- Spatially-constant mean (focus on the 2nd order properties of the data)
- Spatially-constant nugget variance
- Spatial variance: allow this to vary smoothly over CONUS $\rightarrow$ fit the approximation to a stationary GP model (`sigma_model = "approxGP"`) using a coarse, evenly spaced grid of $K = 50$ knots
- Anisotropy:
  - $\rightarrow$ Direction and magnitude of spatial dependence likely to depend on elevation
  - $\rightarrow$ But, may vary across CONUS (Appalachians vs. Rockies vs. Cascades)
  - $\rightarrow$ Components vary linearly with elevation, longitude, interaction (`Sigma_model = "compReg"`)
- Relatively large data set ($N = 2311$): use the SGV likelihood approximation with $k = 15$ neighbors (`likelihood = "SGV"`)

## Markov chain Monte Carlo

- Large number of parameters to sample: 67 total
  - → Mean: 1
  - → Nugget variance: 1
  - → Spatial variance: 50 latent process values + 3 hyperparameters
  - → Anisotropy: 12 (intercept, elevation, longitude, interaction coefficients for each of three sub-processes)
- Univariate adaptive random walk samplers for mean, nugget variance, spatial variance hyperparameters
- Block adaptive random walk samplers for the anisotropy components (quad-variate samplers for each sub-process)
- Need to get creative with the latent process values for $\sigma(\cdot)$:
  - → A single block sampler for all 50 values likely inefficient
  - → 50 univariate samplers will not mix well: parameters likely display strong posterior correlation

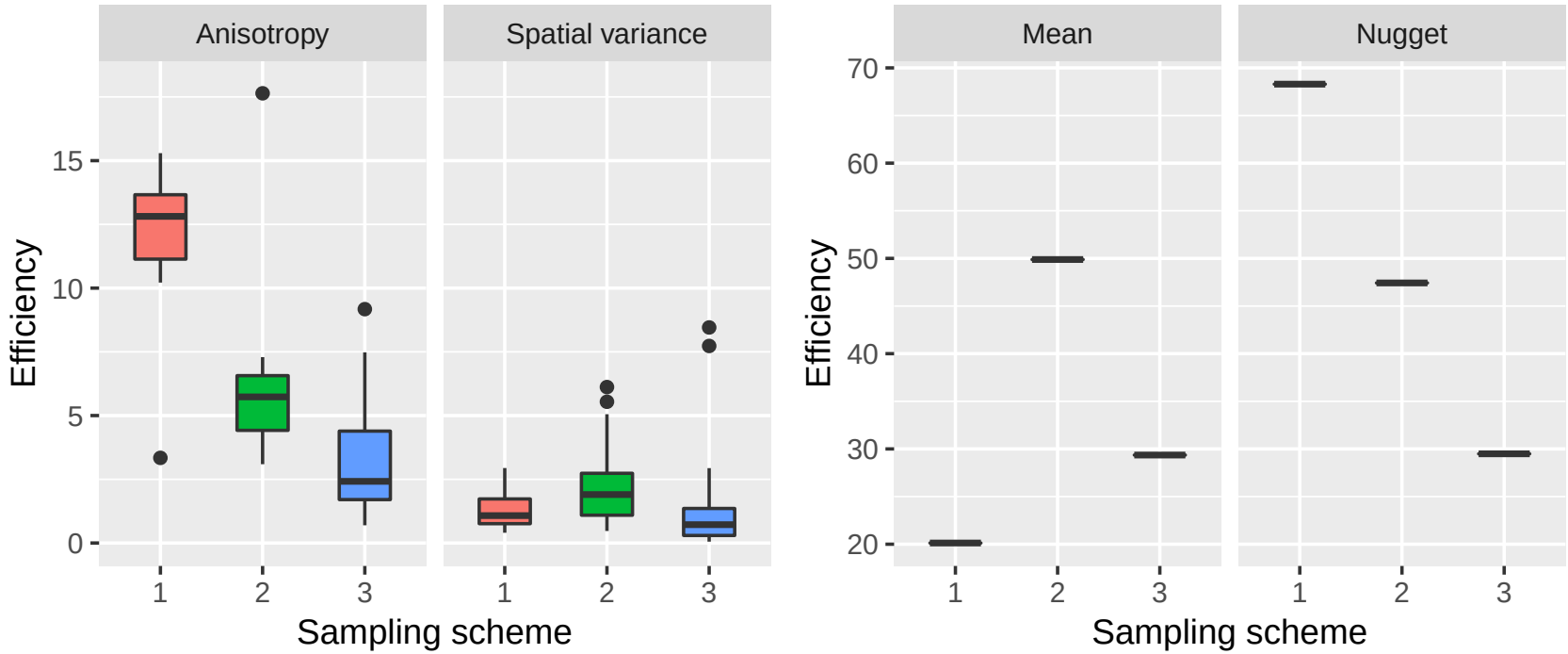
## Markov chain Monte Carlo

Spatial variance knot locations




- Block samplers for latent parameters: three sampling schemes
  1. One large block sampler (50 values)
  2. Five sub-block samplers (9-12 each)
  3. 15 small sub-block samplers (3-4 each)

## MCMC efficiency

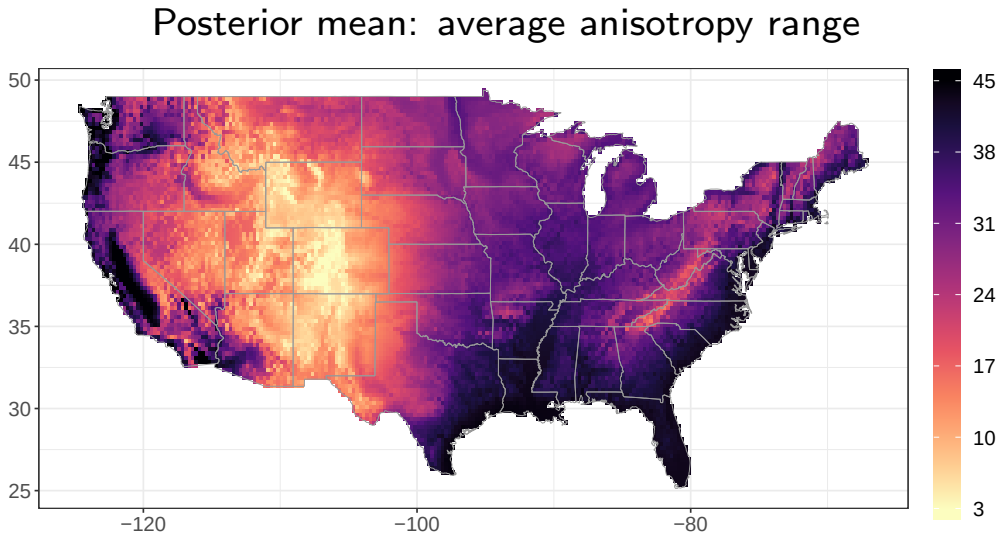



- Efficiency = effective sample size / compute time
- Only changes here are with respect to sampling for the spatial variance: still, a large impact on the other parameters
- Scheme 2 seems to maximize efficiency across all parameters

## Summarizing nonstationarity: anisotropy

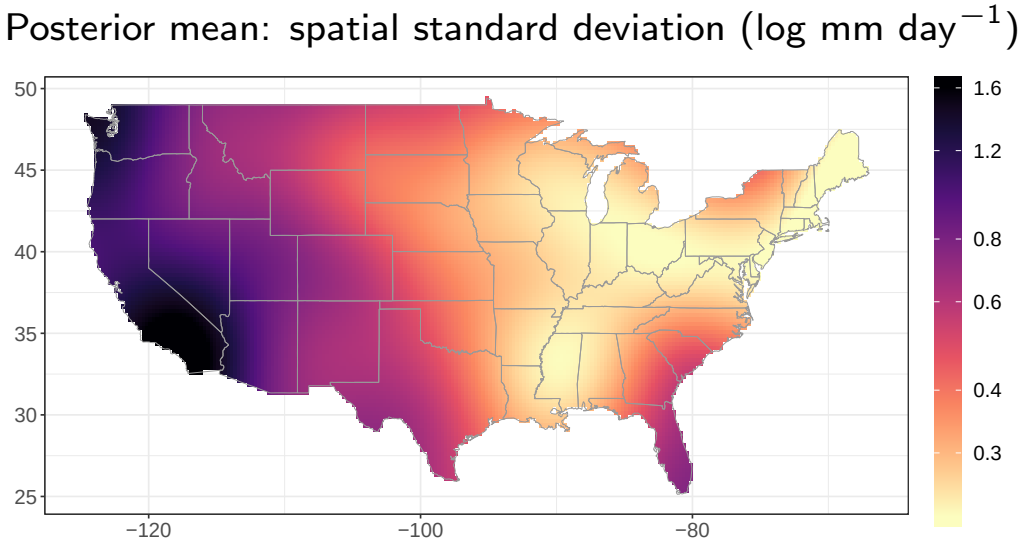

- Recall: summarize variability in magnitude/direction of spatial dependence in terms of elevation, longitude, interaction
- Elevation coefficients are negative $\rightarrow$ higher elevations = shorter range of spatial dependence
- Interaction significantly non-zero in the $x$-direction $\rightarrow$ relationship between elevation and spatial dependence different for western US vs. eastern US

## Summarizing nonstationarity: anisotropy




- Another way to visualize: spatial distribution of the average anisotropy "range" (average of the two anisotropy eigenvalues)
- Map is clearly driven by underlying elevation $\rightarrow$ interesting differences along east coast vs. Rocky Mountains

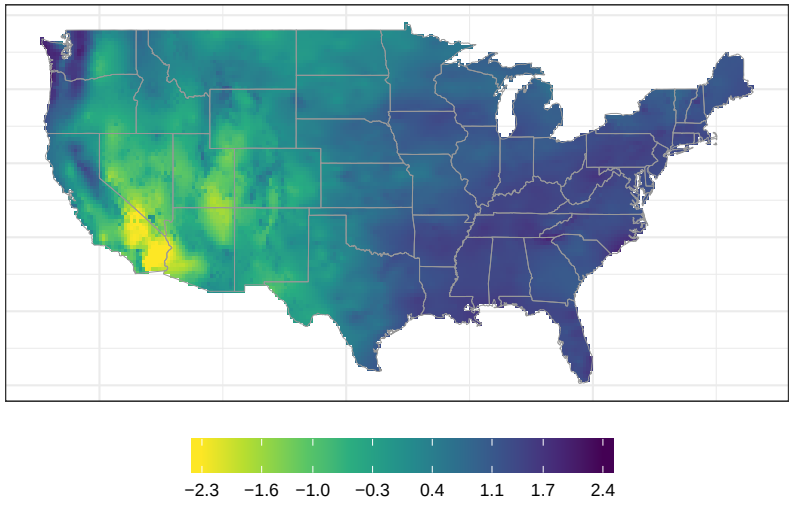
## Summarizing nonstationarity: spatial variance



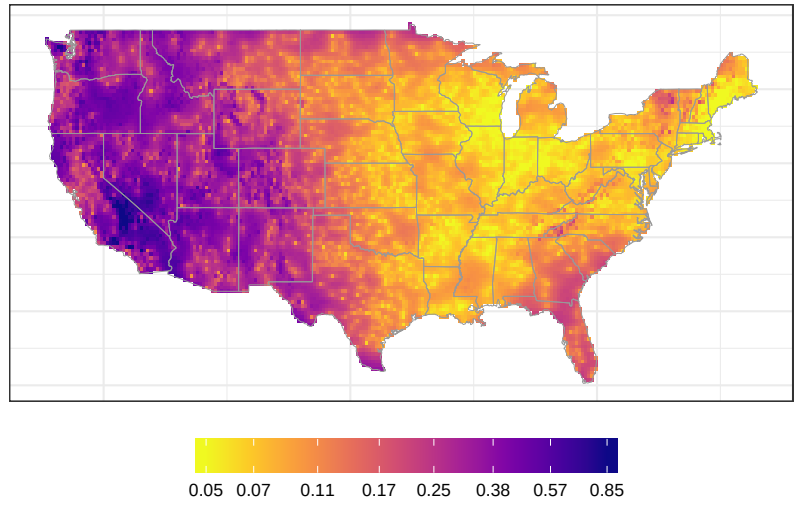

- Map of the approximate GP / nonparametric regression for the spatial variability of precipitation rate
- Extreme variability on the west coast (particularly southern CA); much less variability in the central and eastern US

## Posterior prediction

(a) Posterior mean (log mm per day)

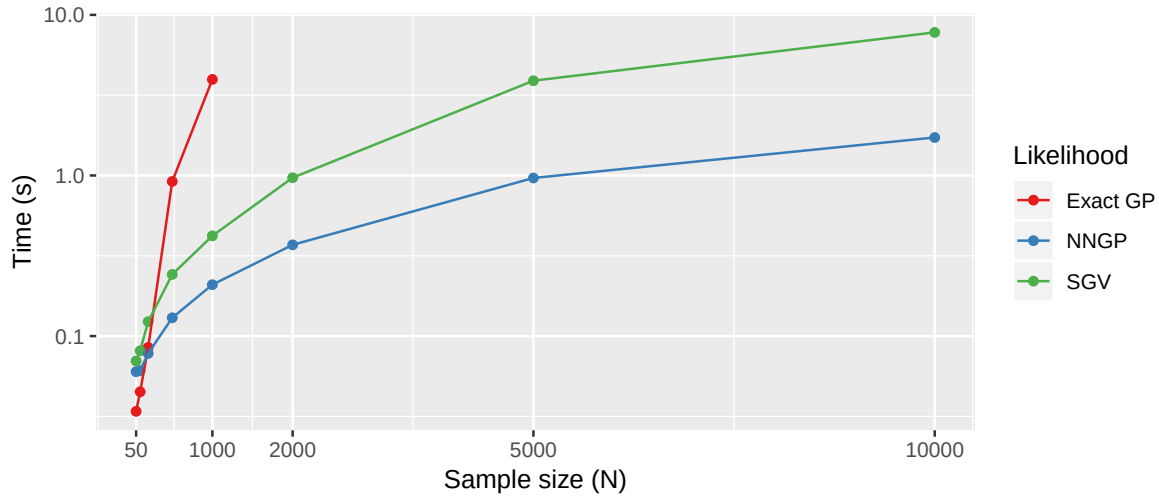



(b) Posterior standard deviation (log mm per day)




- Much clearer picture of the spatial distribution of precipitation rate over CONUS
- Standard errors: station locations visible (uncertainty a function of distance to observed data), but much larger in the western US relative to the central and eastern US

## Computational time: likelihood calculations




- Compute times on a standard laptop
- Approximate methods outperform exact GP for $N > 200$
- NNGP much faster than SGV
- Likely not scalable to $\mathcal{O}(10^6) \rightarrow$ but, enables inference for $\mathcal{O}(10^4)+$ without a custom computing environment

# Thank you!

**Summary**

- Flexible framework for modeling nonstationary spatial processes
- Corresponding software to enable "off-the-shelf" implementation
- Computational tools for implementing flexible Markov chain Monte Carlo
- Approximate GP methods: enable inference for "large" data sets (but not "massive")

Contact: Mark D. Risser, mdrisser@lbl.gov

Paper:

Mark D. Risser & Daniel Turek (2020) Bayesian inference for high-dimensional nonstationary Gaussian processes, Journal of Statistical Computation and Simulation, DOI: 10.1080/00949655.2020.1792472