

# Covariance among parameters

ISTA 410 / INFO 510: Bayesian Modeling and Inference

---

U. of Arizona School of Information

April 12, 2021

Last time:

- Multilevel regression / hierarchical linear models
- Varying intercepts and slopes

Today:

- Modeling covariance among parameters

## Example

---

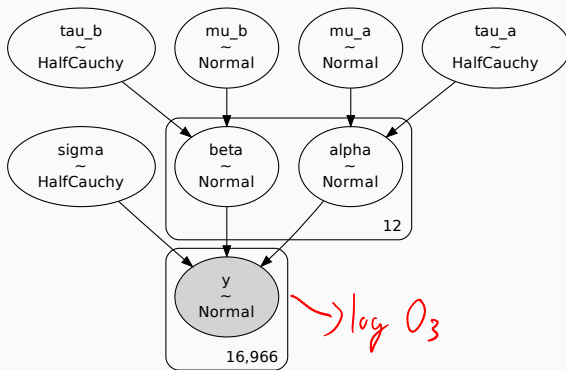
# Beijing air pollution

Example from last time: Data set:  $\sim 5$  years of air quality monitoring data from Beijing

- Weather properties: temperature, pressure, dewpoint, rain, wind speed
- Pollutants: ozone, sulfur dioxide, nitrous oxide, carbon monoxide, particulates
- Measurements collected hourly at 12 monitoring stations

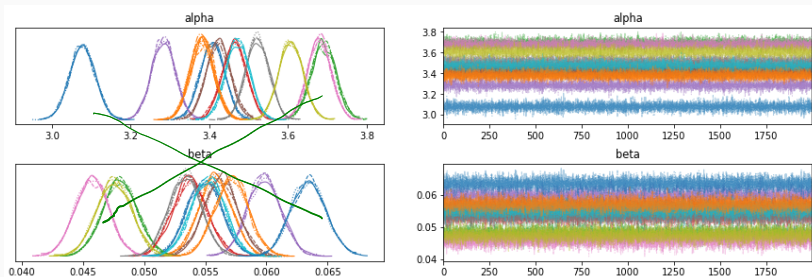
Simple modeling task: model ozone (in  $\mu\text{g}/\text{m}^3$ , about twice ppb) as a function of temperature

# Multilevel model



# Slopes and intercepts

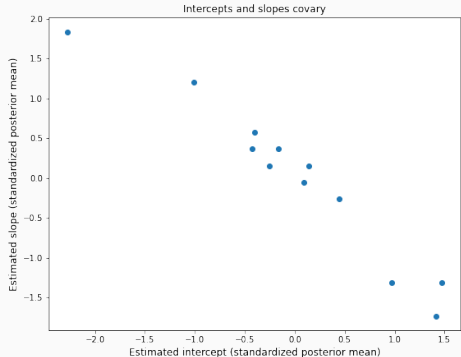
Here is part of the traceplot from the varying-slopes model:



Colors are the same. What do we see?

# Covariance between parameters

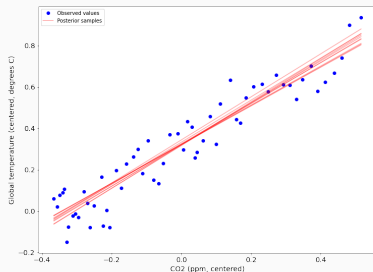
- Intercepts vary across stations
- Slopes vary across stations
- Intercepts and slopes are correlated



A station with a high baseline ozone sees a smaller relative effect with temperature

# Varying slopes and covariance

Covariance across parameter types is common for these models:



Each pair  $\alpha, \beta$  represents a line

- They all have to stay close to the data
- If you change the slope you change the intercept



# The model

varying ids      varying slopes.

$$\log O_3 \sim \text{Normal}(\mu, \sigma)$$
$$\mu = \alpha_j + \beta_j T_{\max}$$

prior for slope-intercept pairs

$$\left\{ \begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} \sim \text{MVNormal} \left( \begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \Sigma \right) \right.$$

$$\Sigma = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} R \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

new - covariance matrix

$$\mu_\alpha \sim \text{Normal}(0, 2)$$

$$\mu_\beta \sim \text{Normal}(0, 0.1)$$

$$\sigma_\alpha \sim \text{HalfCauchy}(1)$$

$$\sigma_\beta \sim \text{HalfCauchy}(1)$$

$$\sigma \sim \text{HalfCauchy}(1)$$

new - prior on correlation matrices

$$\{ R \sim \text{LKJ}(2) \}$$

from previous model

hyper-priors for slope and intercept.

# Multivariate normals and covariance matrices

---

# Multivariate normal distribution

Multivariate normal:

- Generalization of the ordinary normal distribution to produce vectors
- Ordinary normal: parameterized by mean and variance
- MV normal: parameterized by mean vector and covariance matrix

$$\text{Normal}(\mu, \sigma^2)$$
$$\text{MVNormal}(\vec{\mu}, \Sigma)$$

*← covariance matrix*

$m$ -by- $m$  covariance matrix: for  $m$  components,  $m$  variables all varying together

- $m$  standard deviations / variances
- $(m^2 - m)/2$  correlations / covariances
- total:  $m(m + 1)/2$  parameters

## Covariance matrix

Covariance: measures how two random variables vary "together"

Covariance of two random variables:

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

*deviation of X from mean*      *deviation of Y from its mean*

What's the covariance of  $X$  with itself?

$$\text{Cov}(X, X) = E[(X - E[X])^2] = \sigma_X^2$$

## Covariance matrix

Covariance matrix of  $X_1, X_2, \dots$ :

$$\Sigma = \begin{pmatrix} \text{Cov}(X_1, X_1) & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_3) & \dots \\ \text{Cov}(X_2, X_1) & \text{Cov}(X_2, X_2) & \text{Cov}(X_2, X_3) & \dots \\ \dots & & & \end{pmatrix}$$

$$\Sigma_{i,j} = \text{Cov}(X_i, X_j)$$

$$\Sigma_{i,i} = \sigma_i^2$$

$$\Sigma_{i,j} = \Sigma_{j,i}$$

# Covariance matrix

We can write this in terms of variances and correlations:

$$\begin{pmatrix} \boxed{\sigma_1^2} & \sigma_1\sigma_2\rho_{12} & \boxed{\sigma_1\sigma_3\rho_{13}} & \dots \\ \sigma_1\sigma_2\rho_{12} & \boxed{\sigma_2^2} & \sigma_2\sigma_3\rho_{23} & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

variance of  $X_1$

covariance of  $X_1, X_3 = \text{sd}(X_1)\text{sd}(X_3) \cdot \text{cor}(X_1, X_3)$

variance of  $X_2$

$\sigma_1 \cdot \sigma_1 \rho_{11}$

$\rho = \text{"rho"}$   
 $= \text{correlation coefficient}$

## Example

Say  $\sigma_1^2 = 4$ ,  $\sigma_2^2 = 1$ ,  $\rho_{12} = -0.8$ ; then the covariance matrix is

$$\Sigma = \begin{pmatrix} 4 & -1.6 \\ -1.6 & 1 \end{pmatrix}$$

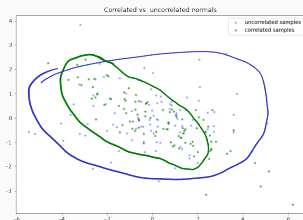
2 · 1 · -0.8

What does a sample from  $\text{MVNormal}(0, \Sigma)$  look like?



## Example

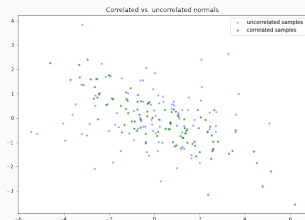
Compare to two independent normals with the same mean, variance:



What would the covariance matrix be for the independent samples?

## Example

Compare to two independent normals with the same mean, variance:

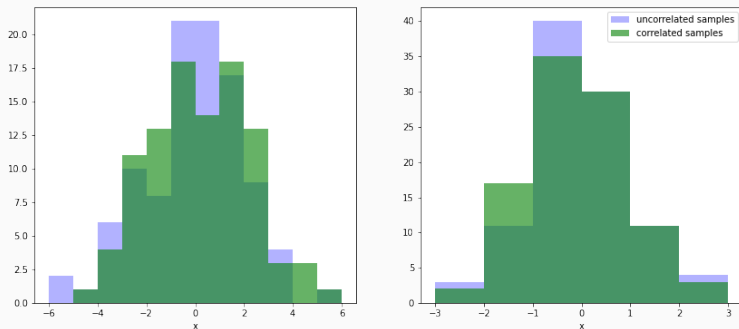


What would the covariance matrix be for the independent samples?

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$$

## Example

Compare the marginal distributions:



Can't tell apart! Marginals are  $N(0, 2)$  and  $N(0, 1)$  in both cases.

## Priors for covariance matrices

---

# Wishart distribution

Previously common: Wishart/inverse-Wishart distribution

Why not to use it:

- Often doesn't reflect our prior beliefs about correlations
- Can't decouple correlation and variance of individual variances
- Computationally hard for MCMC samplers
- Conjugacy not that big a deal anyway

# Factoring the covariance matrix

Matrix multiplication interlude:

$$\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots \\ b_{21} & b_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} = ?$$

# Matrix multiplication

Grab 1st row and 1st column:

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \end{pmatrix}}_{\text{get rows from here}} \underbrace{\begin{pmatrix} b_{11} & b_{12} & \dots \\ b_{21} & b_{22} & \dots \\ \vdots & \vdots & \end{pmatrix}}_{\text{get columns from here}} = ?$$

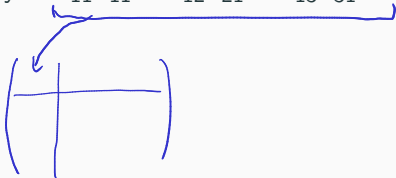
# Matrix multiplication

Grab 1st row and 1st column:

$$\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots \\ b_{21} & b_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} = ?$$

Multiply componentwise and add:

$$\text{product entry} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + \dots$$





# Matrix multiplication

2x2 example:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \end{pmatrix}$$

# Matrix multiplication

2x2 example:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\ Aa + Bc \end{pmatrix}$$

# Matrix multiplication

2x2 example:

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} Aa + Bc & Ab + Bd \\ Ca + Dc & Cb + Dd \end{pmatrix}$$

## $2 \times 2$ covariance matrix

We can factor the covariance matrix:

$$\underbrace{\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}}_{\text{s.d. matrix}} \underbrace{\begin{pmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{pmatrix}}_{\text{correlation matrix}} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

## $2 \times 2$ covariance matrix

We can factor the covariance matrix:

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \underbrace{\begin{pmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{pmatrix}} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

So we have two parts:

- diagonal matrix of standard deviations
- correlation matrix

Priors:

- Standard deviations are easy (Exp, HC, etc.)
- Correlations are a little trickier

## Why do we need a fancy prior?

Need a prior for the correlation matrix  $R$ , and we can't just pick anything:

- Say we have 3 variables,  $x, y, z$
- 3 pairwise correlations:
  - $\rho_{xy} = 0.9$
  - $\rho_{xz} = 0.8$
  - $\rho_{yz} = -0.9$
- What's wrong?

## Why do we need a fancy prior?

Need a prior for the correlation matrix  $R$ , and we can't just pick anything:

- Say we have 3 variables,  $x, y, z$
- 3 pairwise correlations:
  - $\rho_{xy} = 0.9$
  - $\rho_{xz} = 0.8$
  - $\rho_{yz} = -0.9$
- What's wrong?
- Not every matrix can be a correlation / covariance matrix!

# LKJ distribution

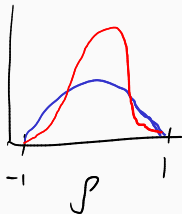
LKJ distribution: a prior for correlation matrices.

- Named for Lewandowski, Kurowicka, and Joe (2009)
- Depends on a shape parameter  $\eta$ , which specifies difference from zero correlations
  - $\eta = 1$ : uniform correlation matrices
  - $\eta > 1$ : suppress big correlations
  - $\eta < 1$ : prefer big correlations
- PDF:

$$p(R|\eta) \propto (\det R)^{\eta-1}$$

- In our simple case:

$$p(\rho|\eta = 2) \propto 1 - \rho^2$$





# LKJ distribution in PyMC3

```
import theano.tensor as tt
with pm.Model() as cov_model:
    mu_a = pm.Normal('mu_a', 0, 2)
    mu_b = pm.Normal('mu_b', 0, 0.1)
    sigma = pm.HalfCauchy('sigma', 1)

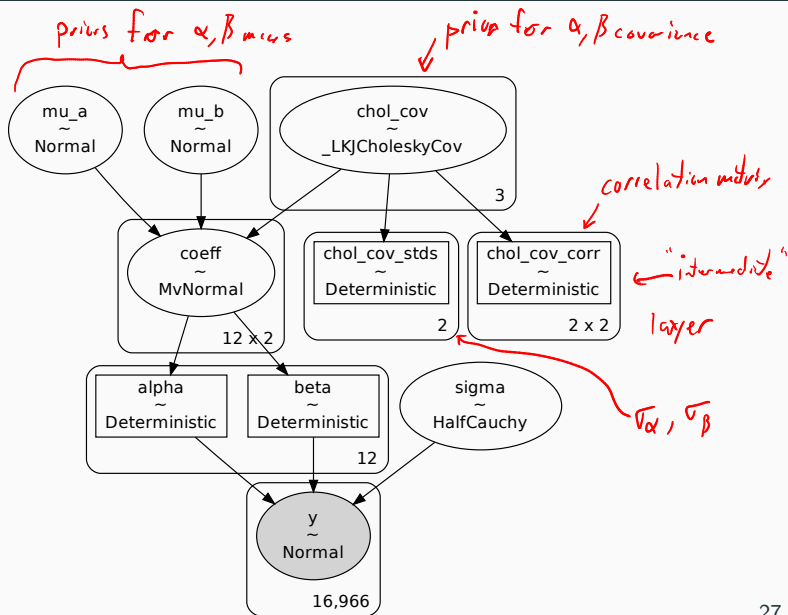
    sd_dist = pm.HalfCauchy.dist(1)
    chol, corr, sd = pm.LKJCholeskyCov('chol_cov', eta = 2,
        n = 2, sd_dist = sd_dist, compute_corr = True)
    # tt.stack makes a vector; chol holds the covariance matrix
    coeff = pm.MvNormal('coeff', mu = tt.stack([mu_a, mu_b]),
        chol=chol, shape=(12, 2))
    alpha = pm.Deterministic('alpha', coeff[:, 0])
    beta = pm.Deterministic('beta', coeff[:, 1])

    theta = alpha[data['station_id']] +
        beta[data['station_id']] * data['TEMP']
    y_ = pm.Normal('y', mu=theta, sigma = sigma, observed = data['log_ozone'])
```

*Handwritten notes:*

- "bore" distribution* (pointing to `sd_dist`)
- LKJ shape* (pointing to `eta = 2`)
- new stuff* (bracketed around the LKJ distribution and `coeff`)
- explicitly compute correlation* (pointing to `compute_corr = True`)
- only size* (pointing to `n = 2`)
- MvNormal* (pointing to `coeff`)
- prior* (pointing to `coeff`)
- shape is 2 parameters ( $\alpha, \beta$ ) 12 groups in model* (pointing to `shape=(12, 2)`)

## Plate diagram



When we run this model:

```
Sampling 4 chains for 500 tune and 500 draw iterations (2_000 + 2_000 draws total) took 172 seconds.
The acceptance probability does not match the target. It is 0.9894609716484106, but should be close to 1.
The acceptance probability does not match the target. It is 0.9819508058560595, but should be close to 1.
There were 457 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.085127247008543, but should be close to 1.
The acceptance probability does not match the target. It is 0.9893641325448185, but should be close to 1.
The rhat statistic is larger than 1.4 for some parameters. The sampler did not converge.
The estimated number of effective samples is smaller than 200 for some parameters.
```

- Ugly stuff!
- But, we've run into this problem with hierarchical models before, and we know some tricks

# Non-centered parameterization

Remember when we did the 8 schools model:

Bad: *school wise mean treatment effect*  
 $\theta \sim \text{Normal}(\mu, \sigma)$

Good:

$$\theta = \mu + \sigma\eta$$

$$\eta \sim \text{Normal}(0, 1)$$

- Essentially, modeling the standardized  $\theta$ s instead of modeling the  $\theta$ s directly:

$$x_i = \mu + \sigma z_i \quad \text{--- } z\text{-score}$$

- Mathematically equivalent, but much easier for the sampler

# Non-centered parameterization

So if we standardized the usual hierarchical normal model using

$$\theta = \mu + \sigma\eta$$

what do we do for the multivariate version?

- $\mu$  is easy – just use the mean vector  $\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}$
- $\eta$  is easy – just use a bivariate standard normal
- $\sigma$  is the square root of the variance, so we should replace it with the square root of the covariance

# Cholesky decomposition

When a matrix is symmetric and positive definite (e.g. a covariance matrix) it can be factored:

$$\Sigma = L(L)^T$$

where  $L$  is a lower triangular matrix (hence  $L$ ).

If  $z$  is a vector of independent standard normals, then  $Lz$  has the distribution of  $\text{MVNormal}(0, \Sigma)$  – exactly what we want!

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} = \begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_j} \end{pmatrix} + L \begin{pmatrix} z_{\alpha,i} \\ z_{\beta,i} \end{pmatrix}$$

Good news: PyMC3 already gives us the Cholesky factor (because it's useful for numerical stability anyway)

## Non-centered parameterization

### Finishing the non-centered parameterization:

```
import theano.tensor as tt
with pm.Model() as cov_model:
    # clipped out same stuff
```

```
chol, corr, sd = pm.LKJCholeskyCov('chol', eta = 2, n = 2,
    sd_dist = sd_dist, compute_corr = True)
```

```
# make standard normals
```

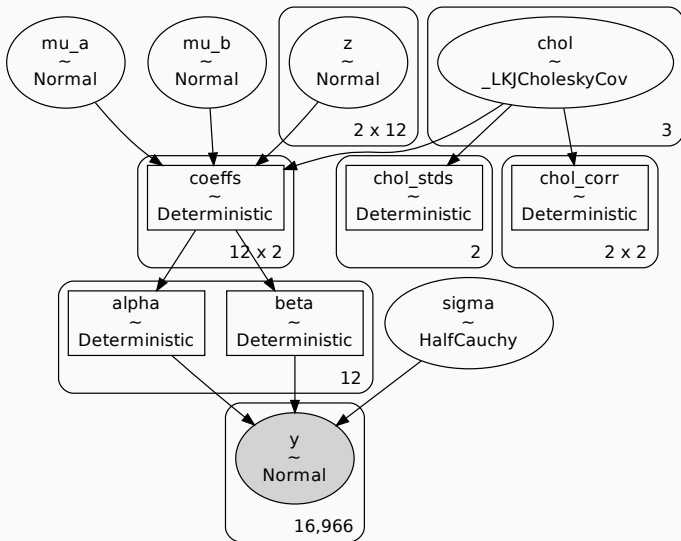
```
z = pm.Normal('z', 0, 1, shape = (2, 12))
```

```
# tt.dot is matrix multiplication
```

```
coffs = pm.Deterministic('coffs',  
                           tt.stack([mu_a, mu_b]) + tt.dot(chol, z).T)
```

```
# clipped out same stuff
```

# Plate diagram

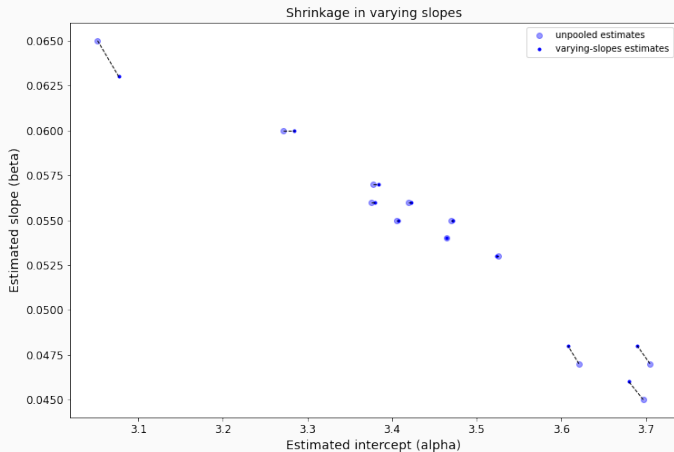




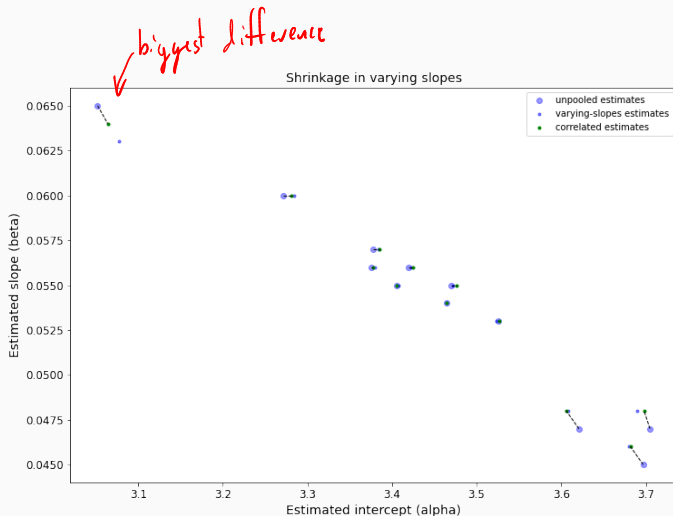
Result: no divergences, good Rhats, sampling takes about 2/3 the time

- $\alpha, \beta$  estimates similar to basic varying-slopes model
- Estimated correlation  $\rho \approx -0.9$
- Two-dimensional shrinkage

# Shrinkage with no correlations



# Shrinkage with no correlations



## What other correlations?

Are there other places we might expect to see correlations in this data set?

- Data come from air quality monitoring stations
- Monitoring stations are located in physical space
- Some are far apart, some are close together

Might reasonably expect that stations located close together are correlated

- Correlation between observations as a function of distance

# Summary

- Using multivariate normal distributions, we can model parameters that are explicitly correlated
- Pool information across different types of parameters as well as different groups
- LKJ distribution gives a regularizing prior on correlation matrices

Next time:

- Spatial correlations and Gaussian processes