A Kalman Filter tutorial

Doug Nychka,

`www.image.ucar.edu/~nychka`

- The state space model
- The KF filter equations
- A normal perspective
- The problems
- Ensembles
- Welcome to the Machine
- Estimating parameters

Institute for Mathematics Applied to Geosciences

Supported by the National Science Foundation

Sandia CSRI workshop, Sante Fe, NM Sep 2007

What these lectures are about.

> The Kalman filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements.

There are many good references on the KF and smoother. A recent one is Randy Eubanks little book with rigorous derivations.

What could I possibly add?

The KF was invented in 1960
What are its seminal ideas?
Where does it break?
What can we do better with 40 years of progress?

The observation and state equations

We want to know where a NASA space capsule is:

- We have irregular noisey observations of it's position
  y_t
- We have a dynamical model to describe how its actual position or state
  x_t
  evolves over time.

An ill-posed inverse problem:

We observe y_t but we really want x_t ...

Observation equation

$$\boldsymbol{y}_t = H\boldsymbol{x}_t + \boldsymbol{e}$$

State equation

$$\boldsymbol{x}_t = G\boldsymbol{x}_{t-1} + u_t$$

with

$$\boldsymbol{e} \sim (0, R) \qquad \boldsymbol{u} \sim (0, Q)$$

H **and** G**,** R **and** Q **are known matrices.**

The KF lives in a linear universe

Some Remarks

Weather Forecasting

Y_t are many different kinds of atmosphere measurements.

x_t is the state of the atmosphere (3-d fields of pressure, temp, water, winds) at time t.

Inherently a sequential problem

Information about the state is accumulated over time, and observations just keep coming ...

Estimation has two basic steps:

1) *(Analysis or Update)* **Update the estimate for the state at time t based on the new observations at time t.**

2) *(Forecast)* **Propagate the state estimate forward to time $t+1$.**

The KF Analysis (or Update)

Prior information from past observations

$$\boldsymbol{x}_t \sim (\boldsymbol{\mu}_t, \Sigma)$$

Kalman update for state

$$\boldsymbol{x}_t^{UP} = \boldsymbol{\mu}_t + \Sigma H^T (H \Sigma H^T + R)^{-1} (\boldsymbol{y} - H \boldsymbol{\mu}_t)$$

Kalman update for covariance

$$\Sigma^{UP} = \Sigma - H^T (H \Sigma H^T + R)^{-1} H$$

What happens tomorrow?

Just propagate the updated mean and covariance forward in time:

$$\boldsymbol{\mu}_{t+1} = G\boldsymbol{x}_t^{UP}$$

$$\boldsymbol{\Sigma} = G\boldsymbol{\Sigma}^{UP}G^T + Q$$

Now ready to update with new observations at time $t+1$.

"Todays forecast becomes tommorrow's prior"

Where does the estimate come from?

Small mean squared error

The KF update is the best unbiased linear estimate of: x_t given y_t and the prior information.

The KF is also the Kriging estimator or optimal interpolation for estimating spatial fields.

Markov property

The update only depends on the current observation and does not depend on the dynamical model directly.

KF catch 22

The exposition of the linear algebra is usually meaningless in a tutorial!

A Normal World

Add Gaussian distribution assumptions

Observation equation

$$\boldsymbol{y}_t = H\boldsymbol{x}_t + \boldsymbol{e}$$

State equation

$$\boldsymbol{x}_t = G\boldsymbol{x}_{t-1} + u_t$$

with

$$\boldsymbol{e} \sim N(0, R) \quad \boldsymbol{u} \sim N(0, Q)$$

(or Prior info from previous updates $\boldsymbol{x}_t \sim N(\boldsymbol{\mu}_t, \Sigma)$)

The filter is the distribution of $\boldsymbol{x}_t$ *given* $\boldsymbol{y}_1, ..., \boldsymbol{y}_t$

This is Gaussian with the mean and covariance from the KF filter!

Sequential updates

If the observations have independent errors then they can be updated sequentially in any order.

This is easy to see from a Bayesian perspective – the likelihood factors provided the observations are conditionally independent given x_t**.**

The KF as a regularization

New data comes in at t the KF updated state estimate is the minimizer of

$$(\boldsymbol{y}_t - H\boldsymbol{x})^T R^{-1} (\boldsymbol{y}_t - H\boldsymbol{x}) \;+\; (\boldsymbol{x} - \boldsymbol{\mu}_t)^T (\Sigma)^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_t)$$

over $\boldsymbol{x}$

This is also proportional to $-log$ posterior density from a Bayesian perspective.

Let's talk about KF problems

High dimensions

Would you like to evaluate the update formula:

$$\Sigma^{UP} = \Sigma - H^T(H\Sigma H^T + R)^{-1}H$$

when Σ has a dimension of 10^5 or 10^6 ?

Solution: Update observation vector sequentially as a scalar and induce many zeroes in Σ

Localization: inducing zeroes

Σ is typically too large to compute with exactly. One approach is a direct multiply by a tapering matrix T

$$\Sigma_{ij}^{TAPER} = \Sigma_{ij} T_{ij}$$

T is positive definite with

$T_{ij} \approx 1$ **For components close to each other**

$T_{ij} = 0$ **For components far away from each other**

Σ^{TAPER} *can be quite sparse*

Substitute Σ^{TAPER} for Σ in the KF equations.

Another big problem

Nonlinear dynamics

Many interesting and practical applications are nonlinear: e.g.

$$\boldsymbol{x}_t = g(x_{t-1})$$

We like the toy system Lorenz '96

$$\frac{dx_j}{dt} = -(x_{j-2})(x_{j-1}) + (x_{j-1})(x_{j+1}) - x_j + F_j$$

$\{x_1(t), ..., x_{40}(t)\}$**: a 40-dimensional system.**

Or how about the atmosphere?

NCAR Community Atmospheric Model

x_t **about** 10^6 **dimensions at 250km resolution.**

Nonlinearity as a problem

Even if the prior is Gaussian and the update is Gaussian, i.e.

$$\boldsymbol{x} \sim N(\boldsymbol{x}^{UP}, \Sigma^{UP})$$

How do we make a forecast using a complex g?

What is the distribution of g($\boldsymbol{x}$)?

This is the mother of all change-of-variables problems!

Solution: Use an ensemble (Monte Carlo) approach to approximate the distribution.

The ensemble Kalman filter (EKF)

The main idea

An ensemble is a *sample* useful for approximating the continuous distribution including covariances among variables.

$$\boldsymbol{x}_t^1, \ldots, \boldsymbol{x}_t^M$$

Approximate any aspect of the distribution using the sample statistics of the ensemble.

e.g. ensemble mean $\approx$ expected value of distribution.

The update and forecast steps just modify each member of the ensemble.

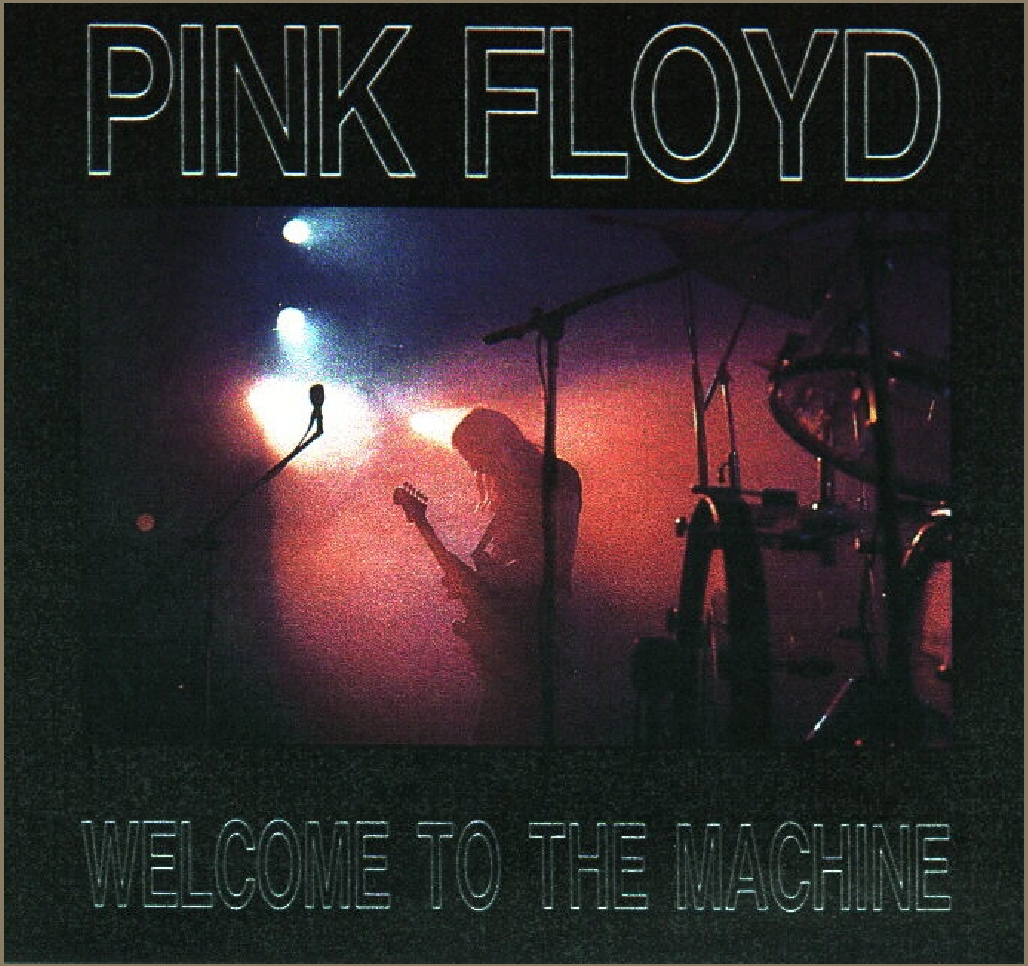
Simple regression: the Machine

By updating observations sequentially the KF can be run by repeatedly using an algorithm based on simple linear regression: (*The Machine*).

The algorithm will be illustrated by a surface ozone spatial data set. This is appropriate because the update step only requires a prior distribution, not the state equation for the dynamics.

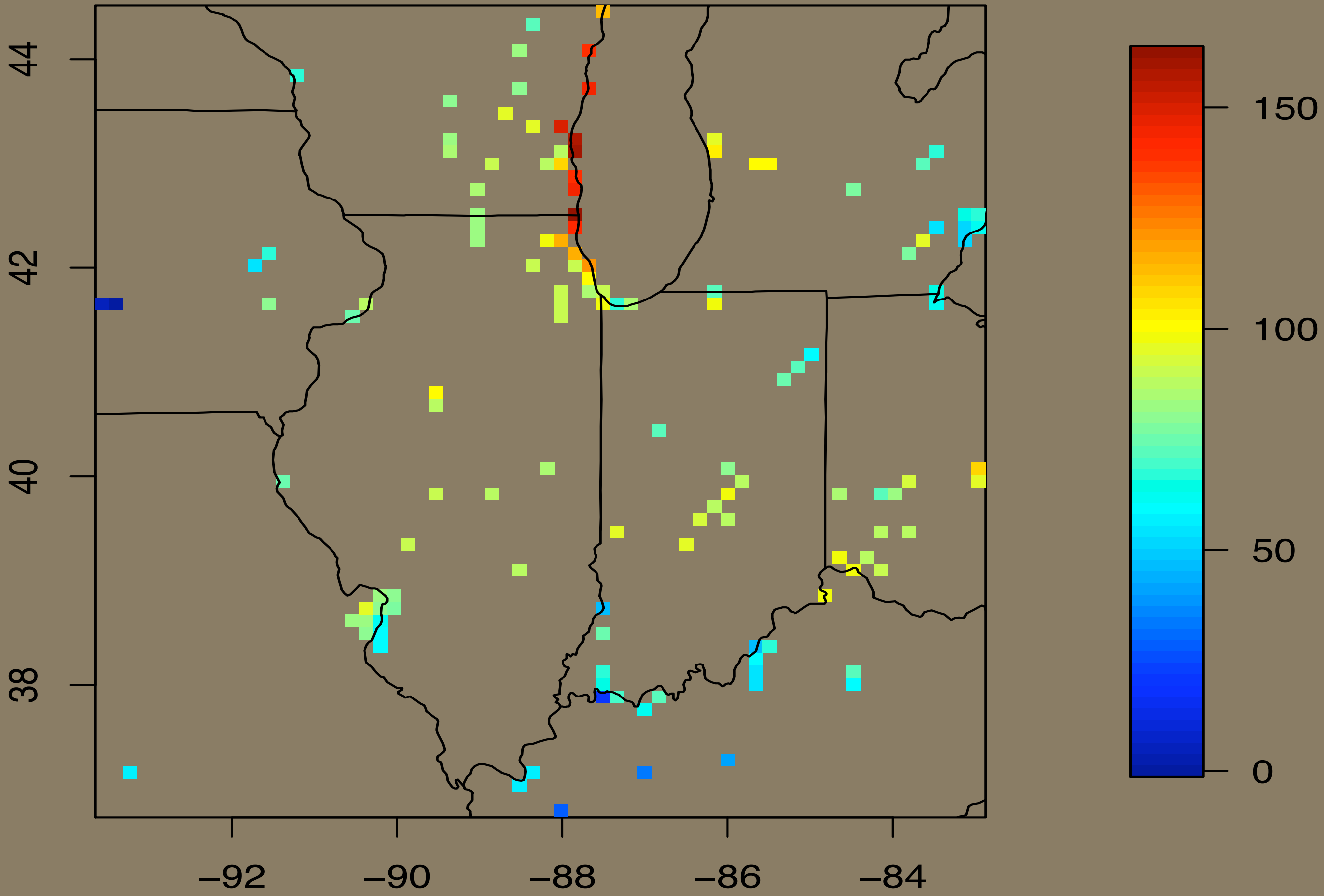
It is of course possible to construct models for the dynamics of ozone fields – but we will not do this here.

Welcome to the Machine



Observed surface ozone, June 19, 1987

Goal: Estimate the surface!



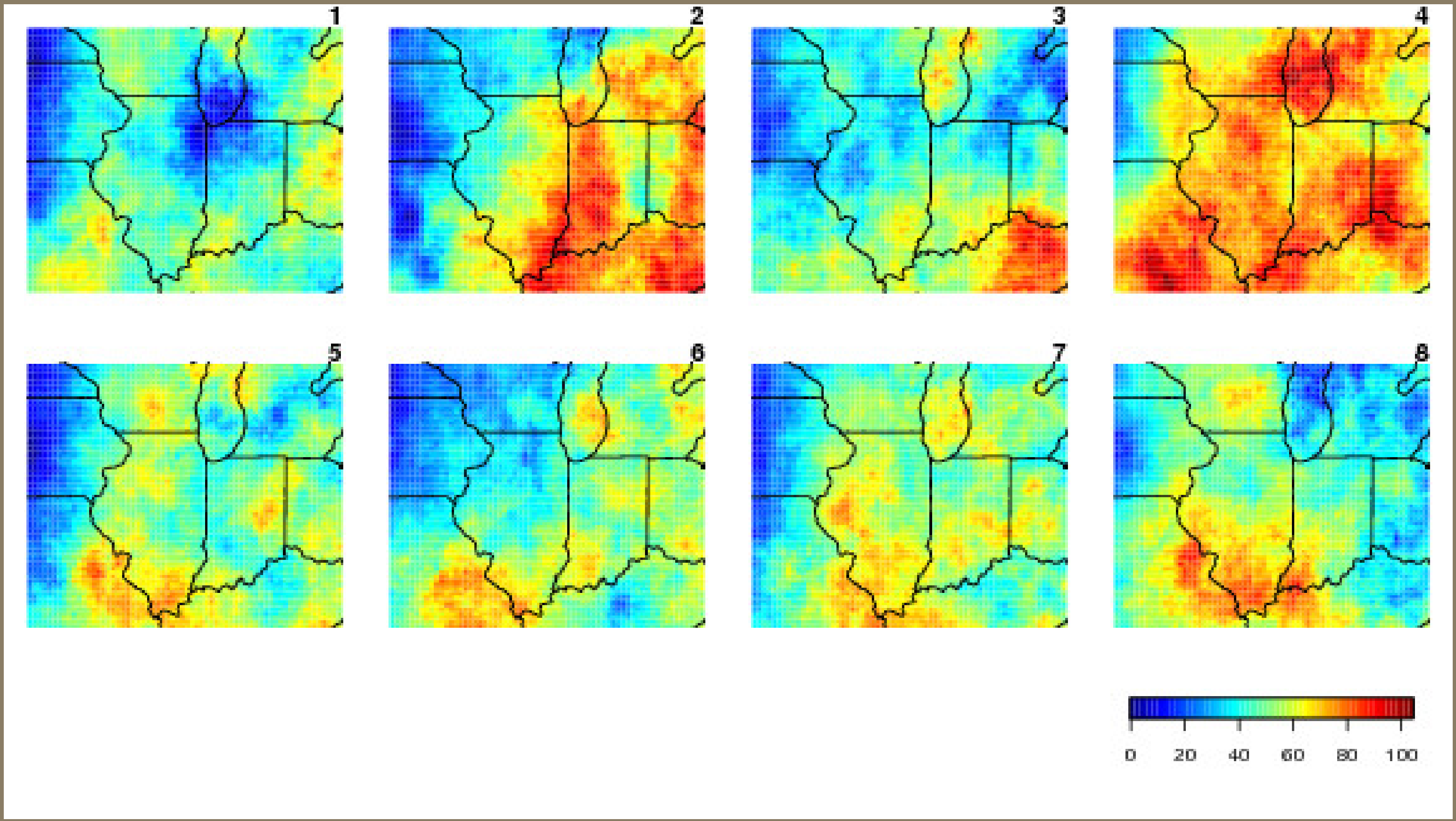
The statistical ingredients for the prior information

Based on data analysis, ozone is (roughly) Gaussian

- **mean around 60PPB a variance from 10 to 25 PPB**
- **a correlation range of about 300 miles.**

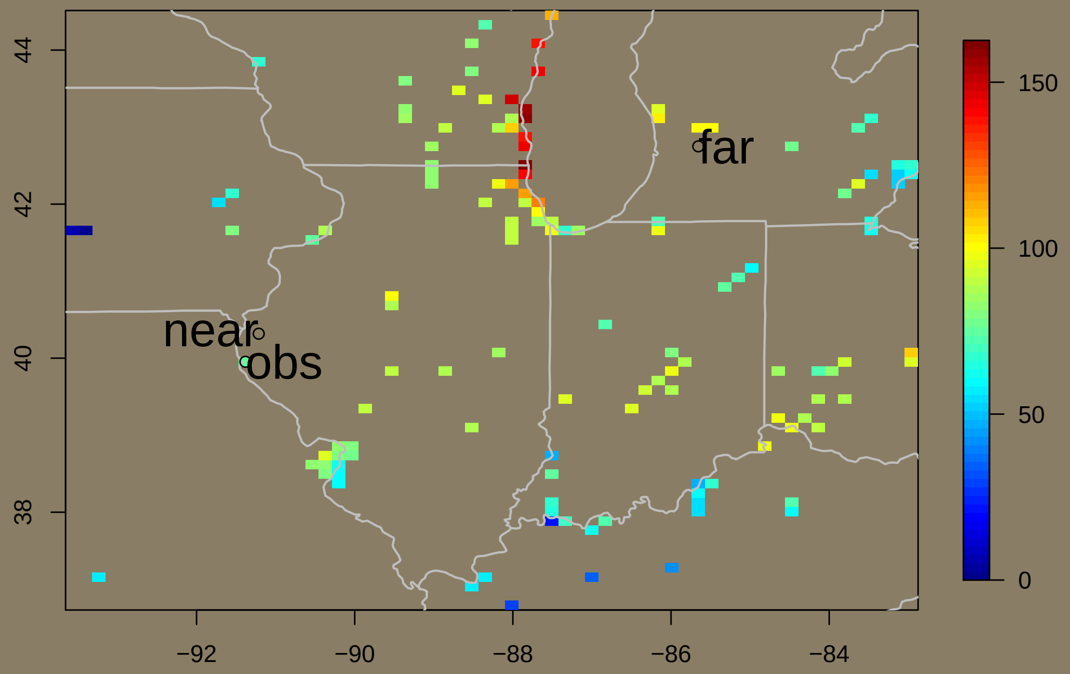
The initial ensemble is 100 random draws from this distribution.

The first 8 members of initial ensemble fields



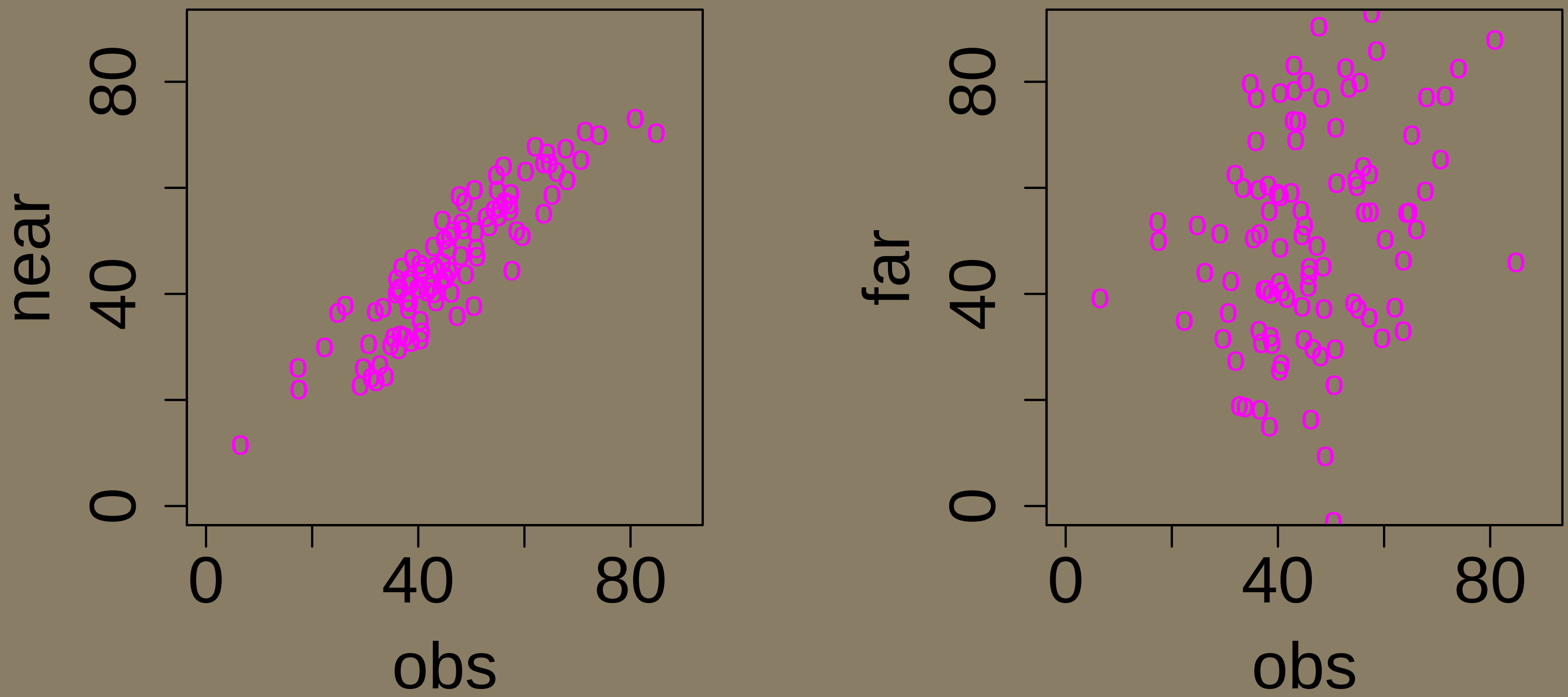
Updating one observation

Observation has value 75 PPB



Consider updates at near and far points.

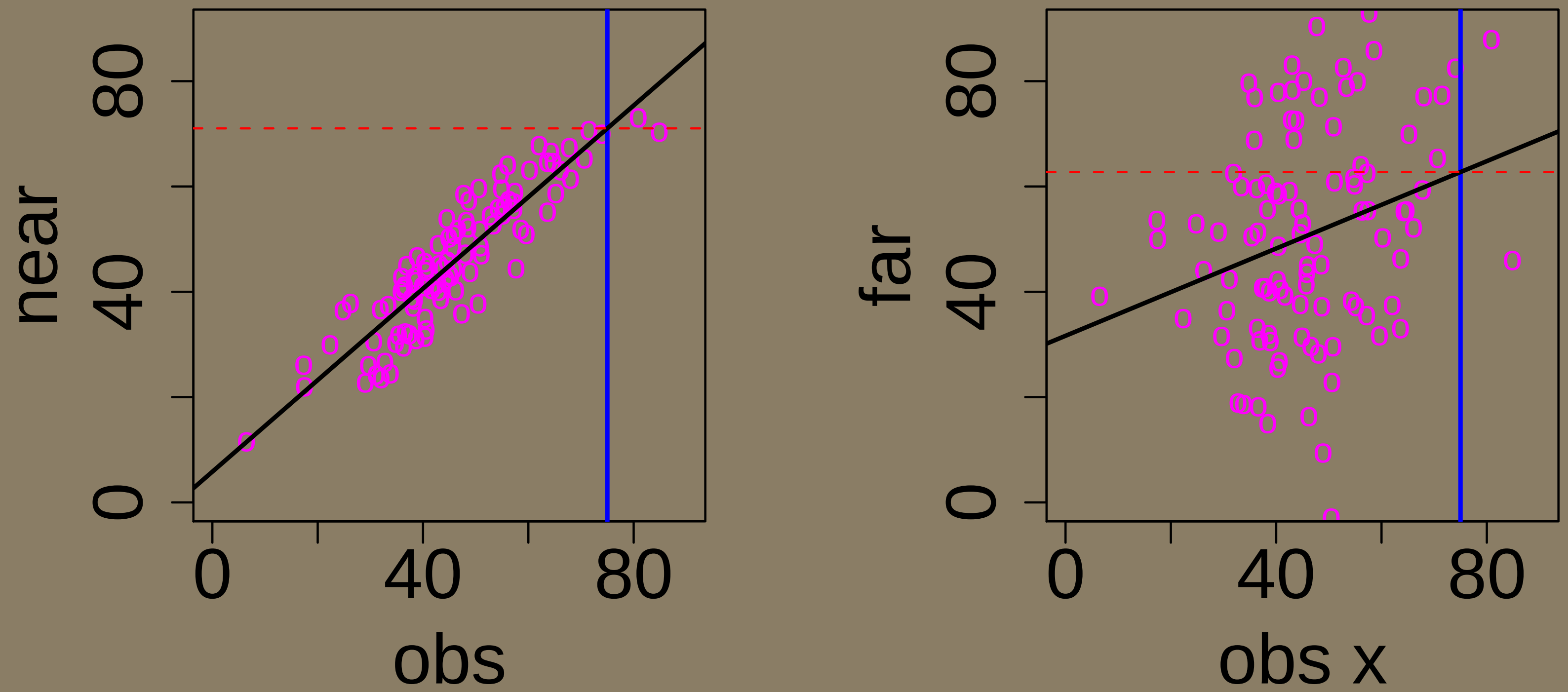
Relationships among the ensemble members



Plots of the pairs of points from 100 ens. members.

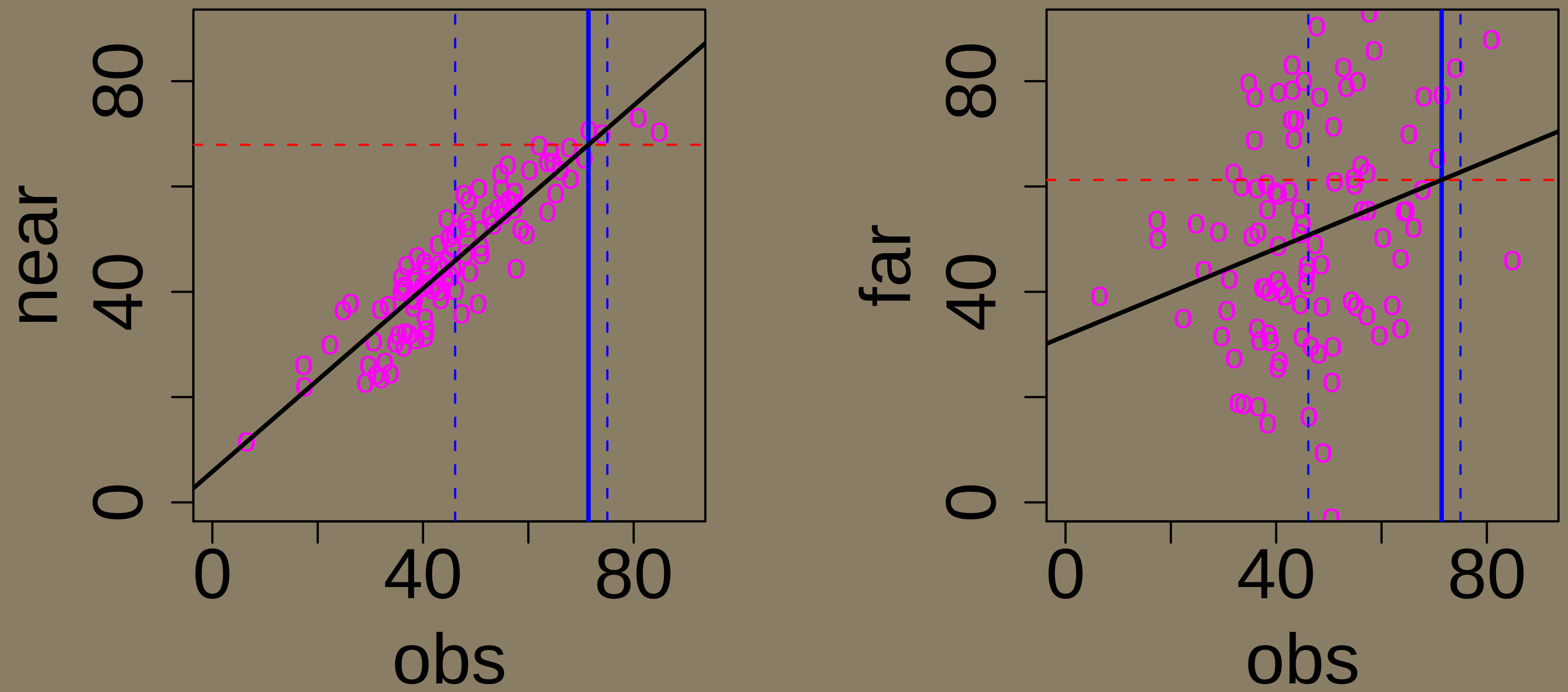
Predicting the grid point from the obs:
Looks like regression!

With no measurement error



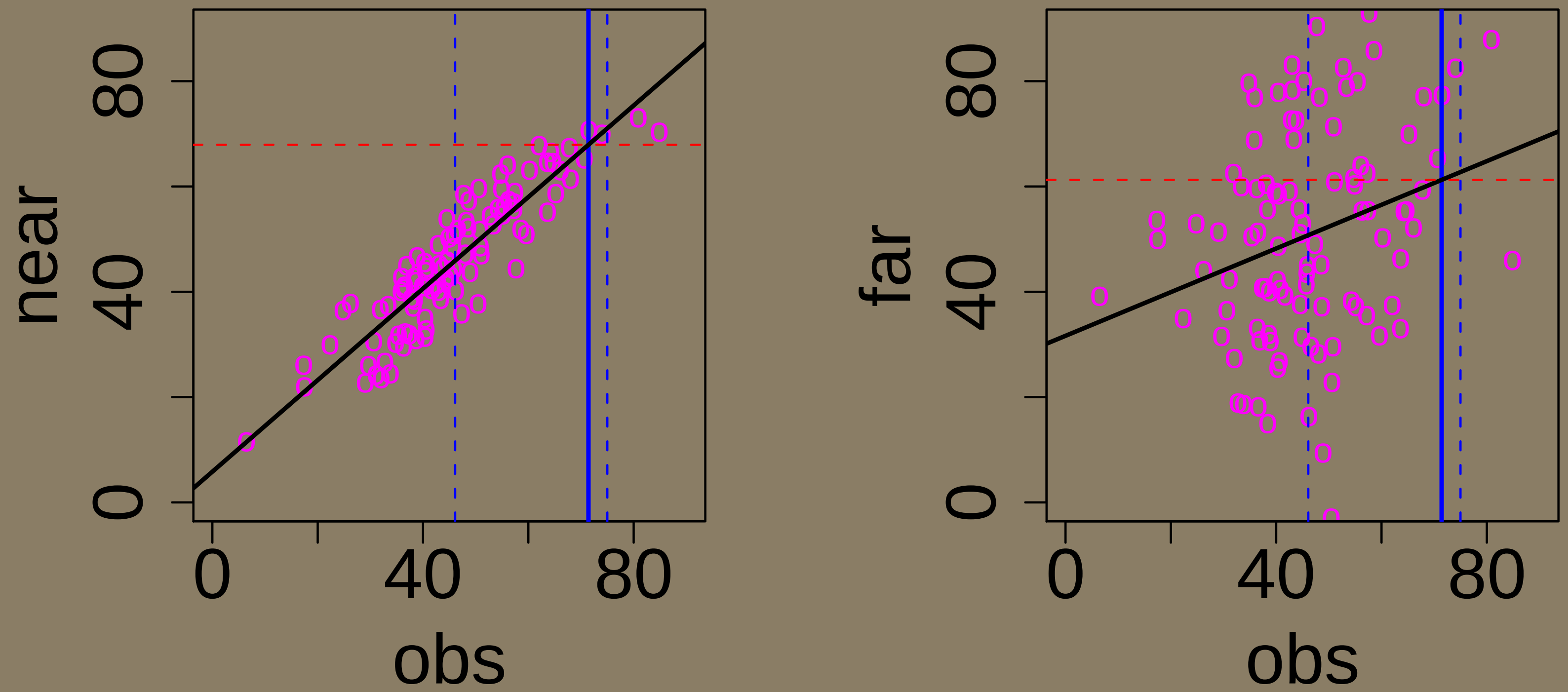
Observation Y=75. **These are least squares lines.**

Adding measurement error



Y= 75 has some error, so adjust for this by shrinking toward the ensemble mean. The Kalman filter tells you how to do this – a weighted average of the prior mean and the observation.

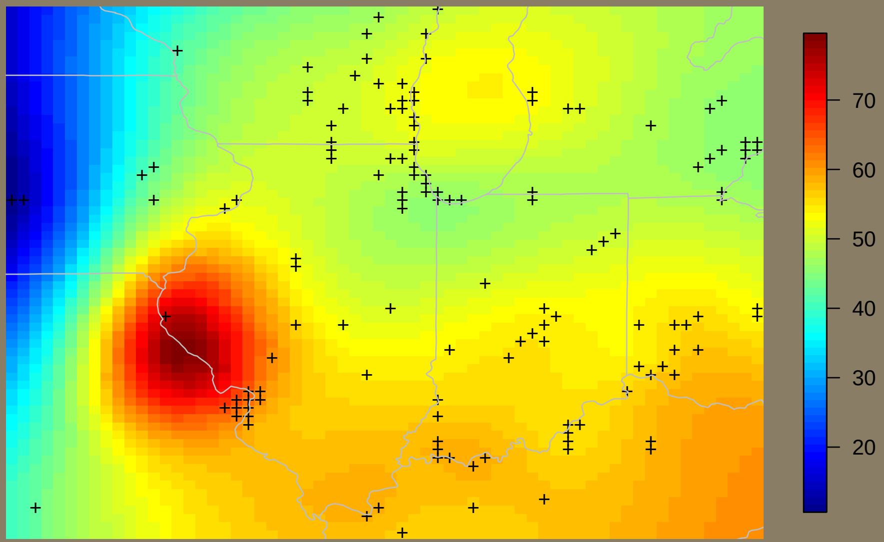
The Machine = g



(up and over)(shrink to mean)[data]

The estimated mean ozone surface

Apply the machine to all grid points.



What is wrong here?

Updating each ensemble member

Add peturbations (or error fields) to the mean estimate.

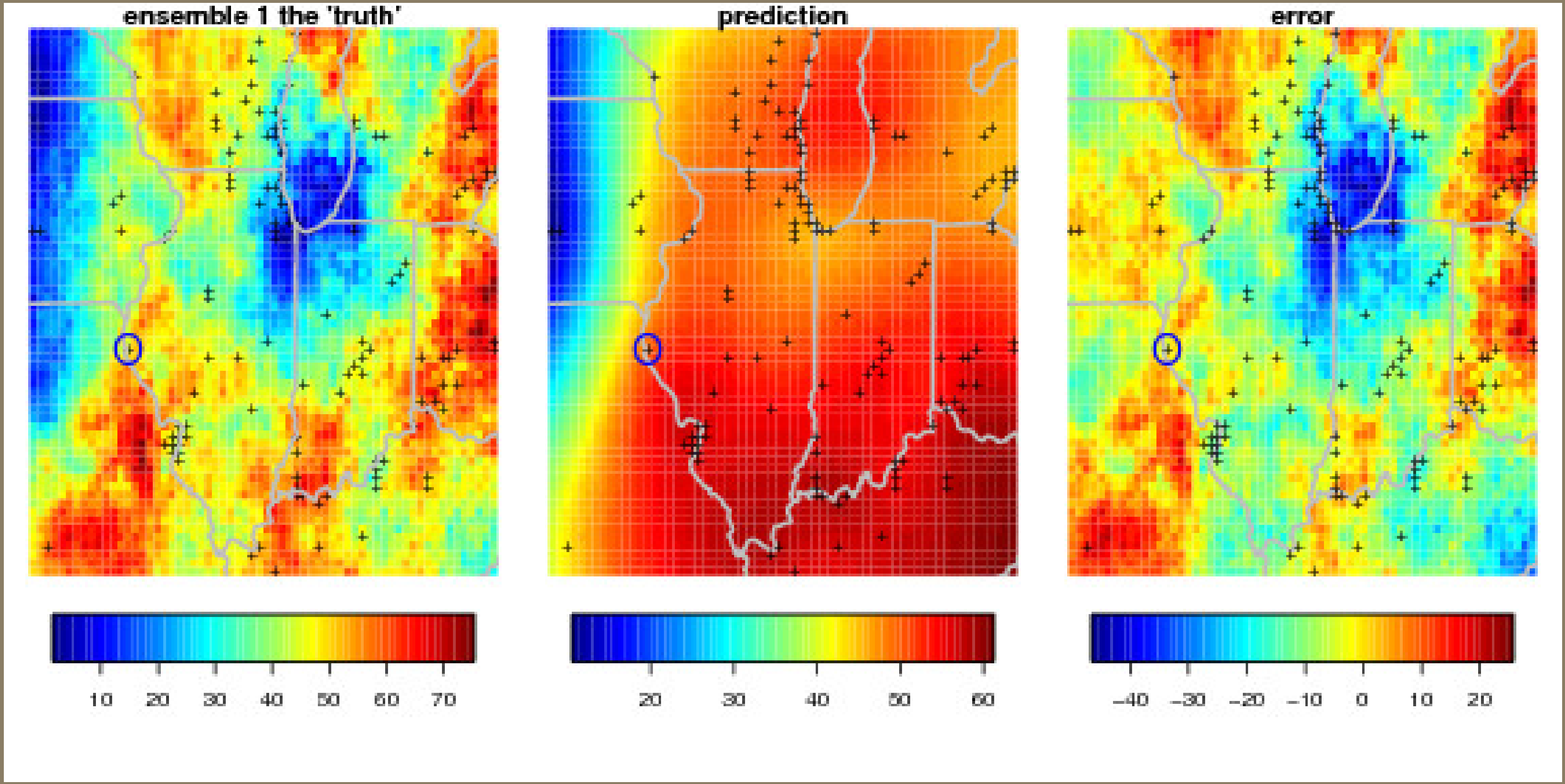
There are several ways to do this but the easiest is just to use Monte Carlo/resampling.

New ensemble member = Mean estimate + error field.

- **Choose an ensemble member (from the prior) and call this "truth".**
- **Generate a pseudo observation at the observation location by adding noise to the ensemble value.**
- **Estimate the field using The Machine.**
- **(estimate - "truth") is a draw from the error distribution.**

Simulating the error field with pictures

Ensemble member, estimated field, prediction errors



An error field from updating the first observation.

What about the forecast step ?

Just apply g to each ensemble member.

This is step is exact!

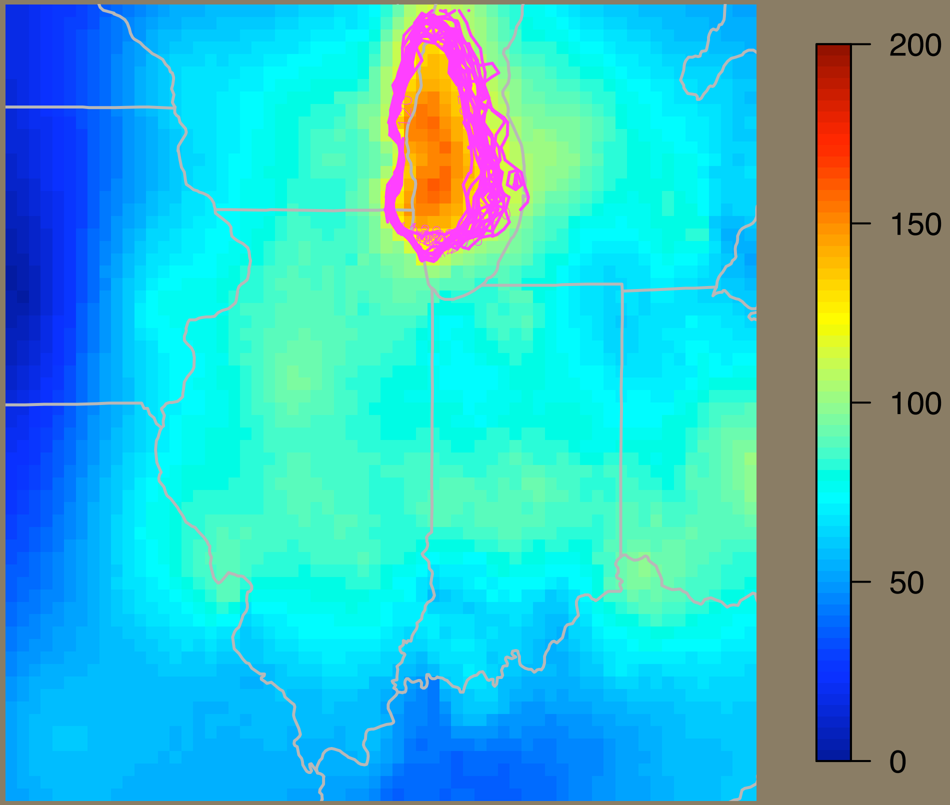
If the ensemble is a draw from the correct distribution then

$$g(x_t^1), g(x_t^2), ..., g(x_t^M)$$

will be a draw from the forecast distribution at $t+1$.

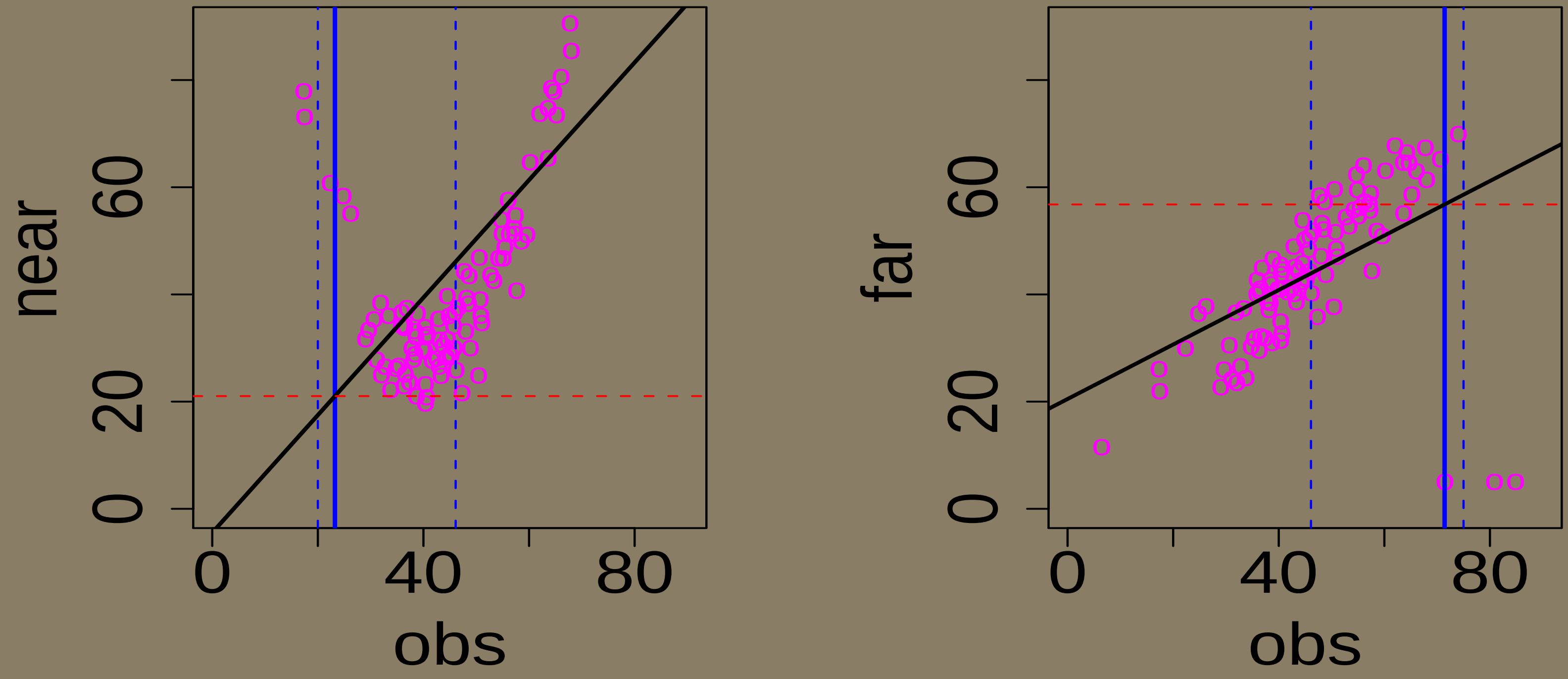
An inference: Where does ozone exceed 120PPB

Find the ensemble contours at 120PPB.



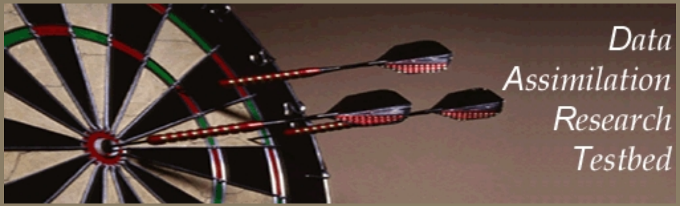
Some Research

Nonlinear relationships **or outliers**



Need a new MACHINE!

Try this in your home or office



DART *Data Assimilation Research Testbed*

Using the MACHINE: Estimating parameters

A full atmospheric climate model is too expensive to run for many different parameter settings.
But many of the parameters need to be tuned ...

- Add the parameters to the state of the system.
- Filter weather observations over time.
- Update both the state and the parameter using an Ensemble Kalman Filter.

An example using Lorenz '96

$$\frac{dx_j}{dt} = -x_{j-2}x_{j-1} + x_{j-1}x_{j+1} - x_j + F_j$$

Suppose the forcings in L'96 are unknown – can they be estimated?

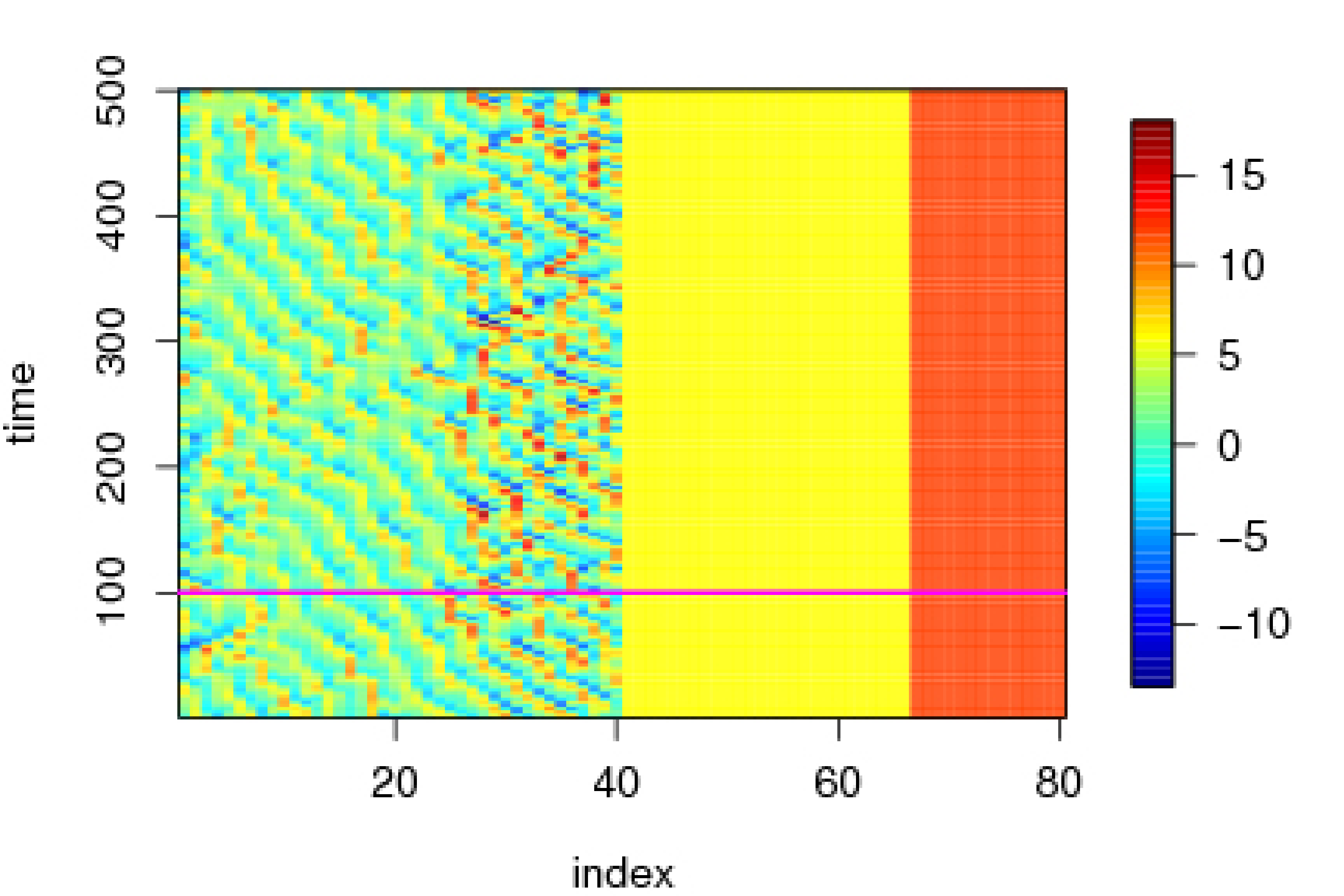
Augment the state vector to include these 40 extra parameters as part of the state

The dynamics for the $\{F_i\}$ is just a random walk.

Proof of concept

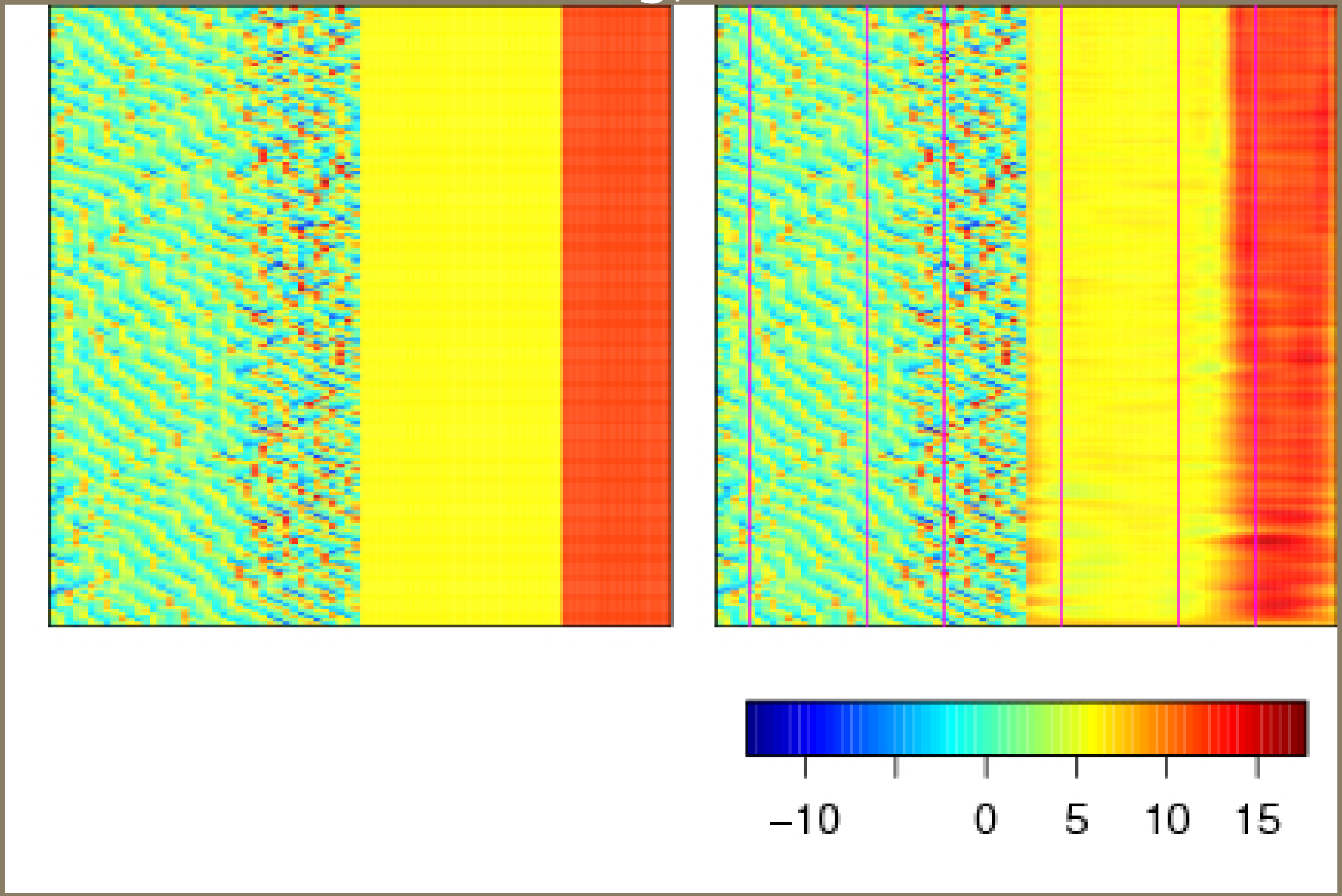
Suppose: $F_1 ... F_{30} = 6$ **and** $F_{31} ... F_{40} = 12$

A peek at the system:



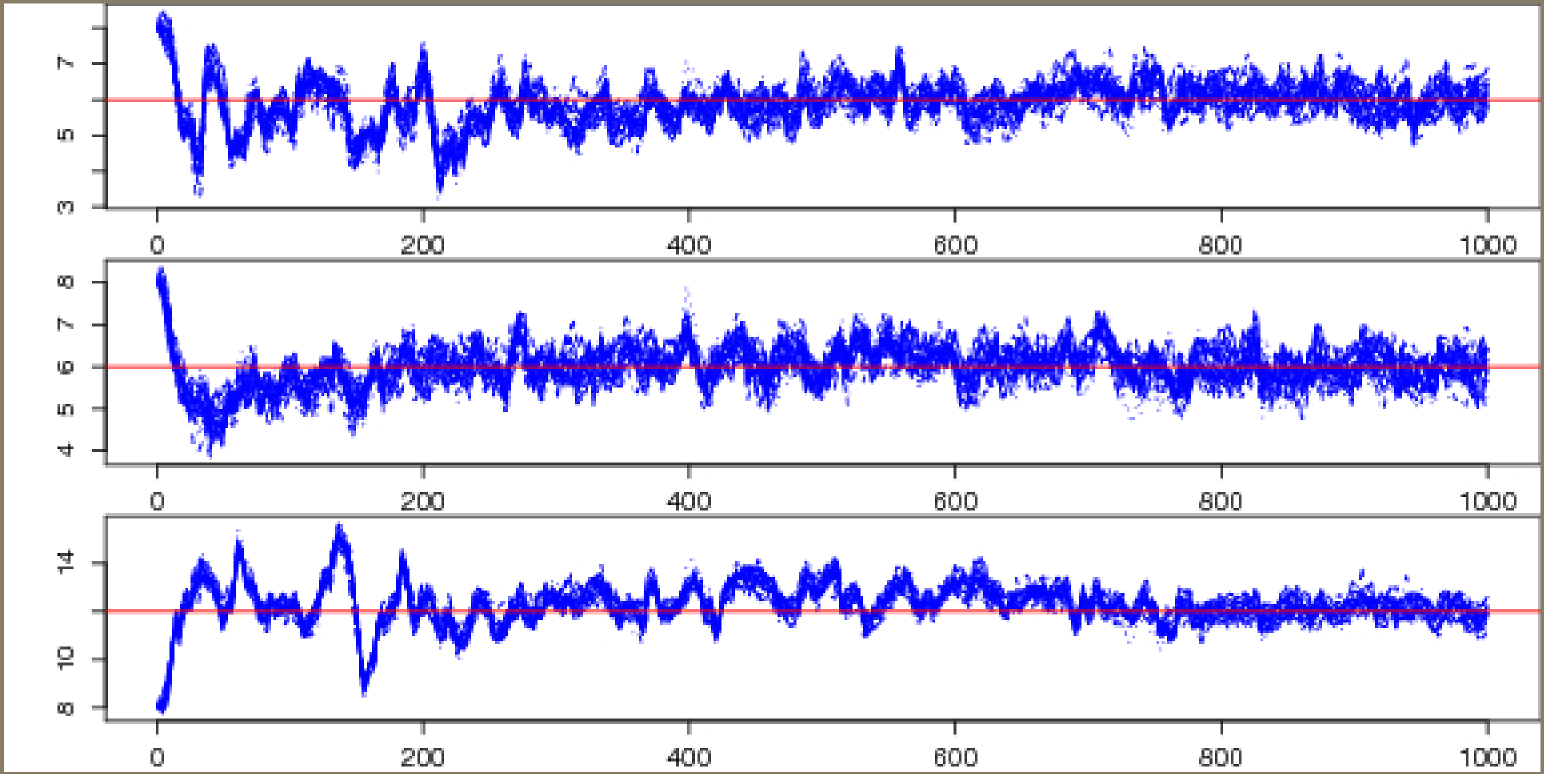
Filter estimates

True states and forcing, Estimated



There is no reason why this should work, But it does!

F at locations 5, 20 and 30.



Summary

- The state space model is an important framework for inverse problems and filtering.
- Pink Floyd has made algorithmic contributions to the KF.
- EKF has potential to estimate some parameters in models.