State-Space Models

State-Space Models

- hierarchical model – data generation and process stochasticity are at different levels
- model variation in the ecological process(es) separately from observation error

State-Space Models

For when you observe this state :

Y_t

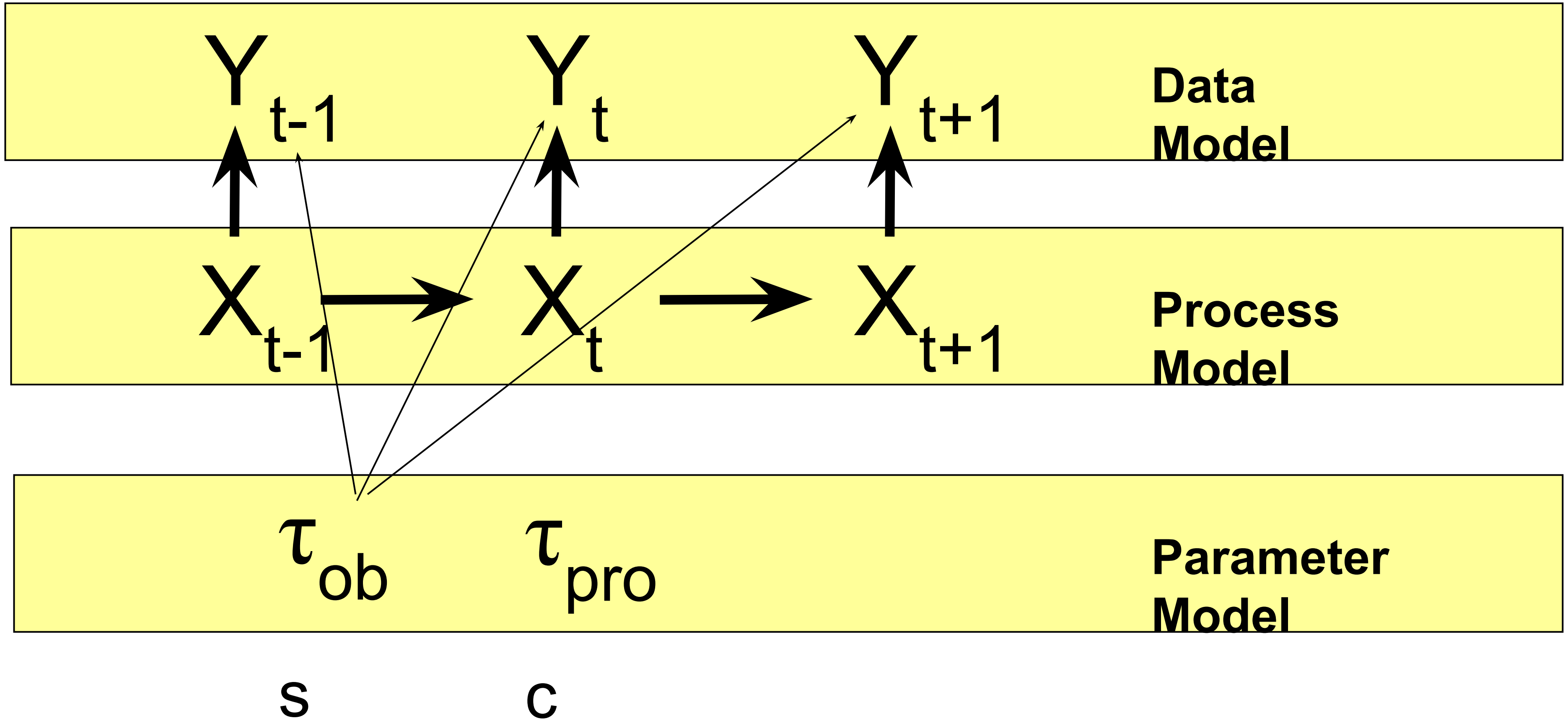
But want to estimate this:

X_t

State-Space Models

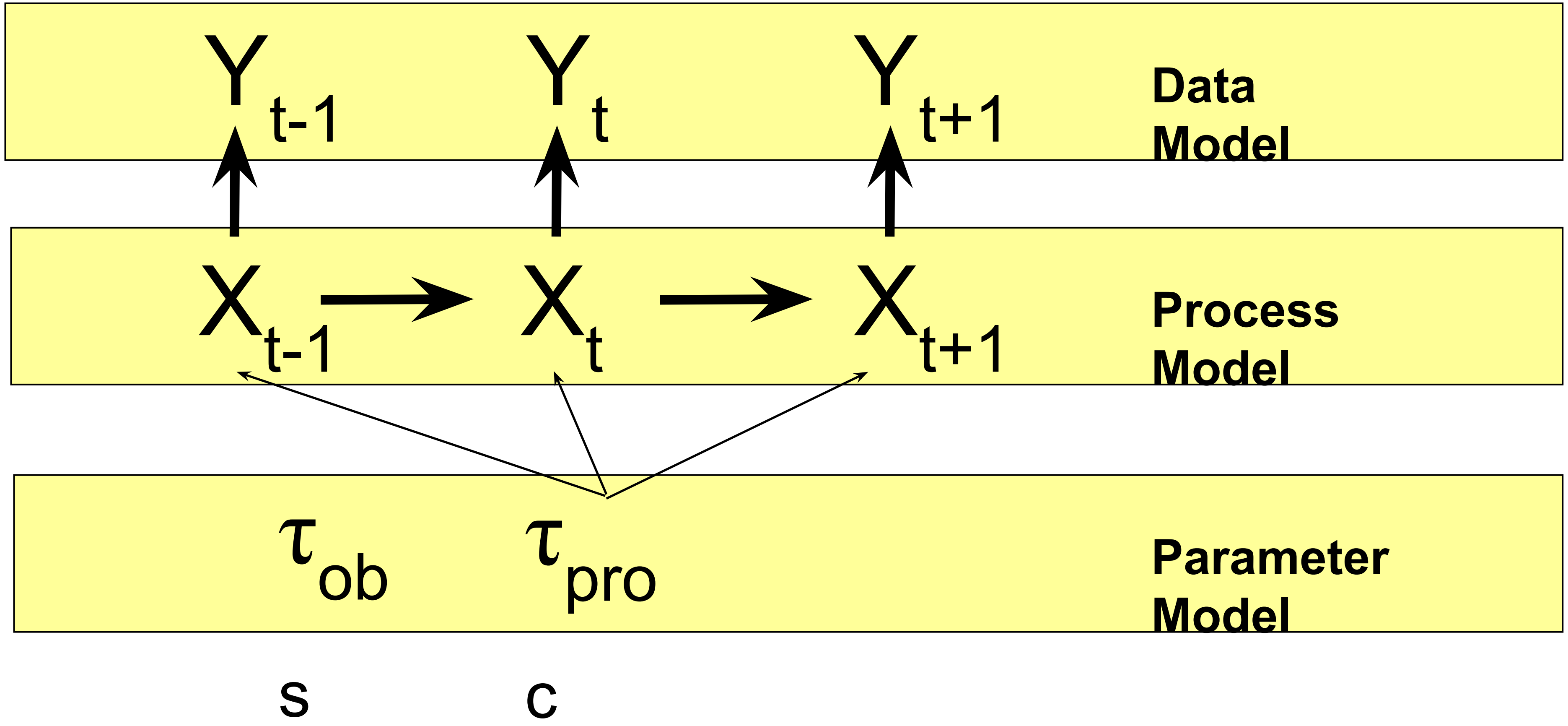
- hierarchical model – data generation and process stochasticity are at different levels
- model variation in the ecological process(es) separately from observation error
- framework commonly used with time-series data
- autocorrelated hidden (true) states informed by independent observations

Random Walk



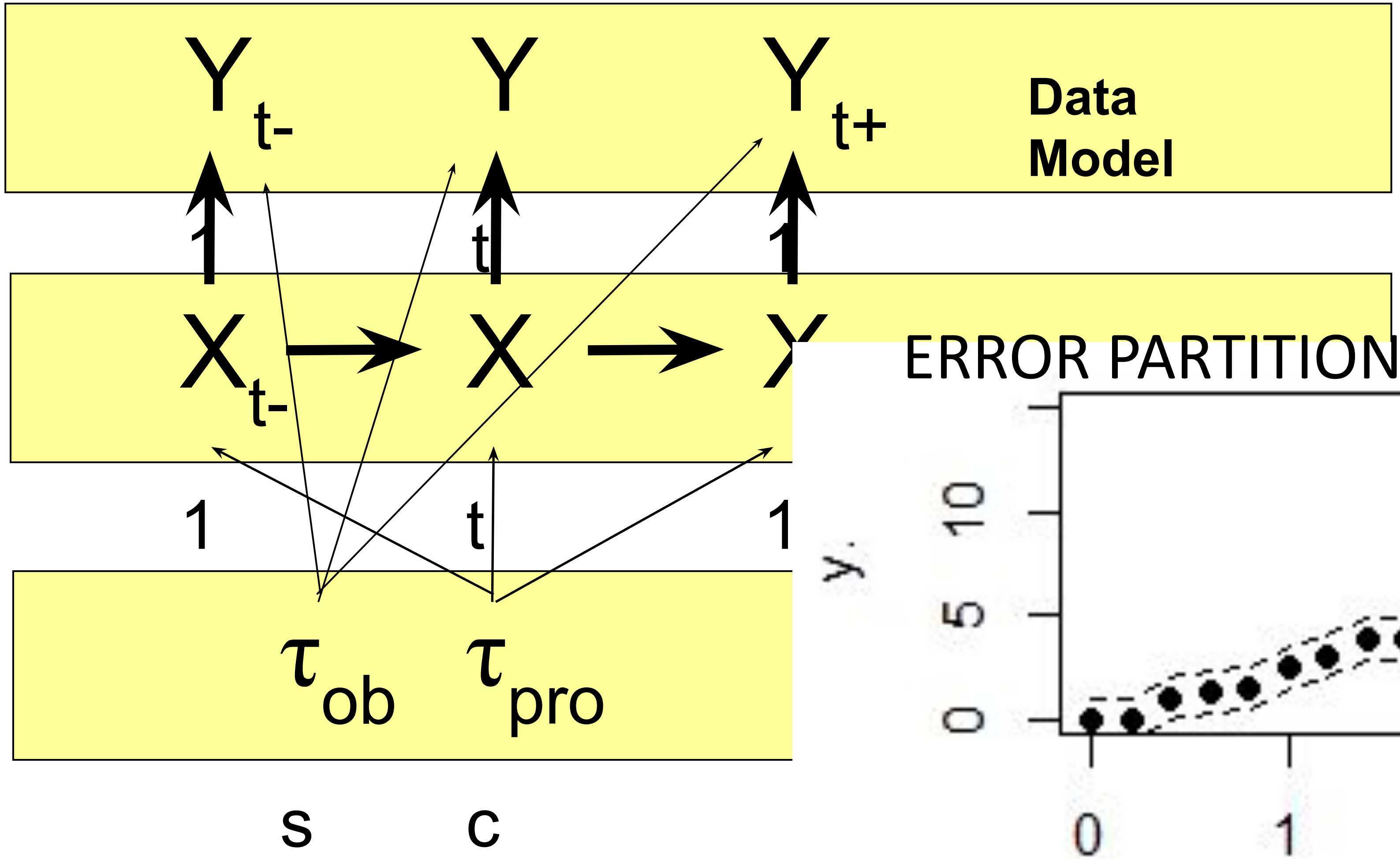
Y's are conditionally independent given the X's

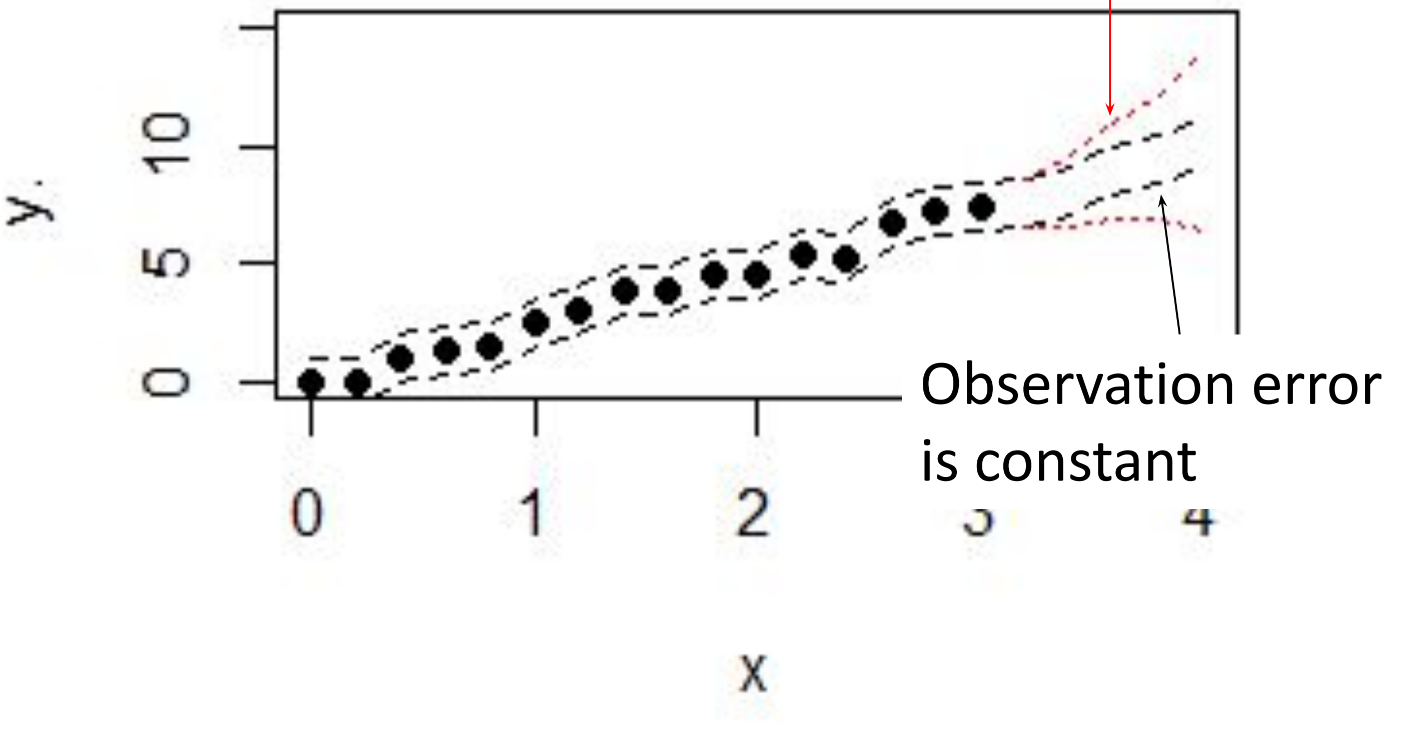
Random Walk State-Space Model



autocorrelation in X: X_t estimate depends on X_{t+1}, X_{t-1} and Y_t

ERROR PARTITIONING





State-Space Models

$$X_t = f(X_{t-1}) + \varepsilon_t$$ **Process Model**

$$Y_t = g(X_t) + \omega_t$$ **Data Model**

- X = <u>latent</u> time series
- Y = observed data
- ε = process error
- ω = observation error

State-Space Models

$$X_t = f(X_{t-1}) + \varepsilon_t$$ **Process Model**

=b0 + b1* X_{t-1} + b2*covariate

$$Y_t = g(X_t) + \omega_t$$ **Data Model**

- X = <u>latent</u> time series
- Y = observed data
- ε = process error
- ω = observation error

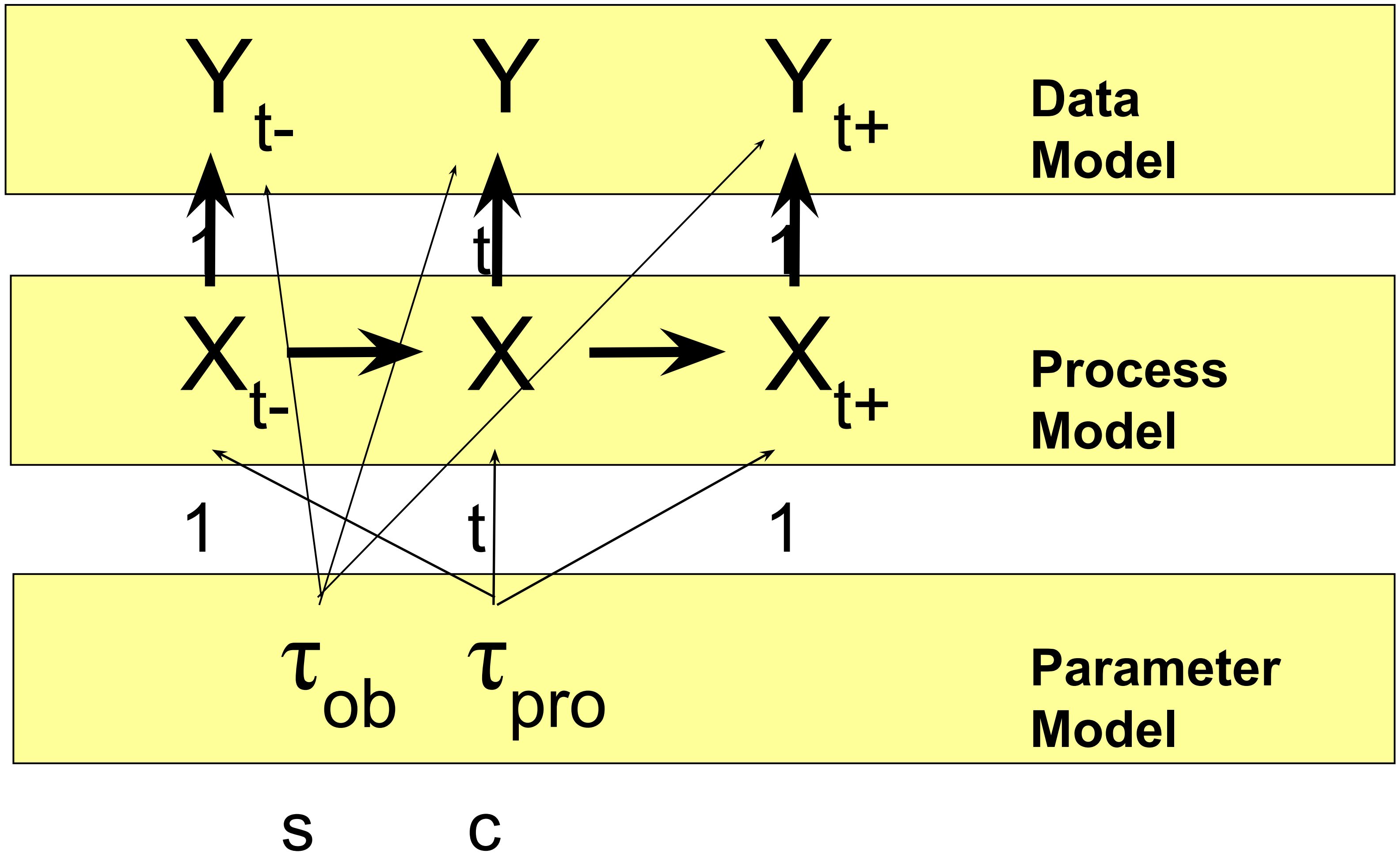
State-Space Models
random walk

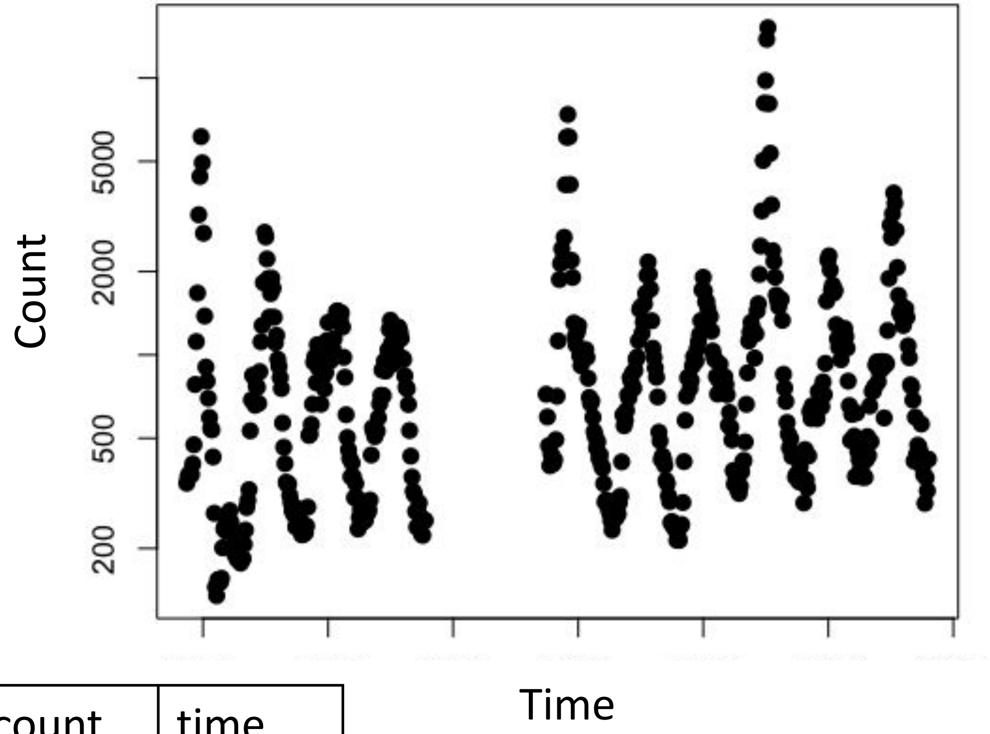
$$X_t = X_{t-1} + \varepsilon_t$$ **Process Model**

$$Y_t = g(X_t) + \omega_t$$ **Data Model**

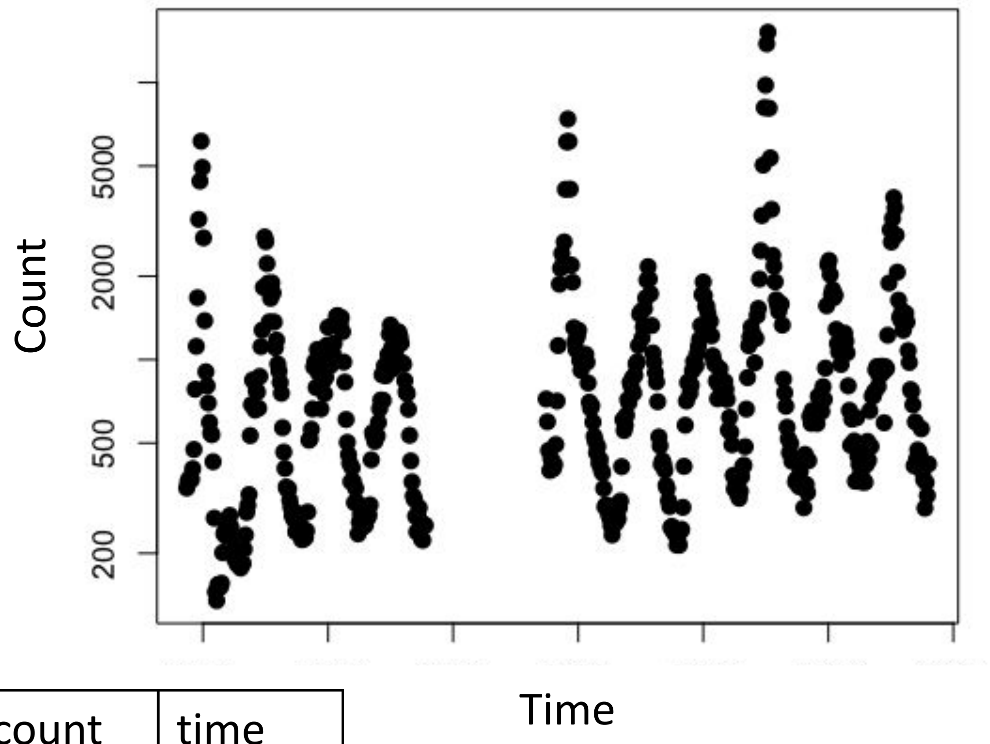
- X = latent time series
- Y = observed data
- e = process error
- w = observation error

```
RandomWalk = "
model{

  #### Data Model
  for(t in 1:n){
    y[t] ~ dnorm(x[t],tau_obs)
  }

  #### Process Model
  for(t in 2:n){
    x[t]~dnorm(x[t-1],tau_proc)
  }

  #### Priors
  x[1] ~ dnorm(x_ic,tau_ic)
  tau_obs ~ dgamma(a_obs,r_obs)
  tau_proc ~ dgamma(a_proc,r_proc)
}
```



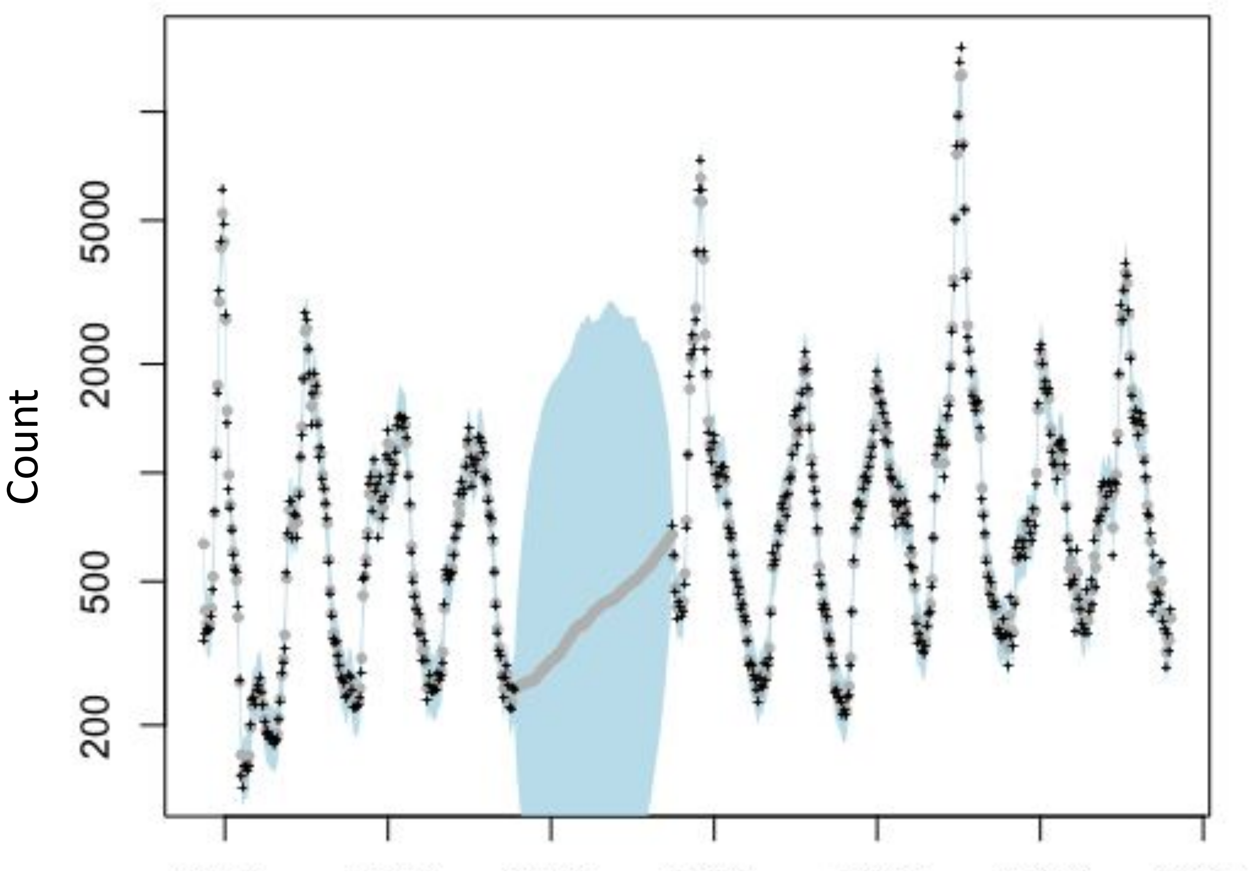


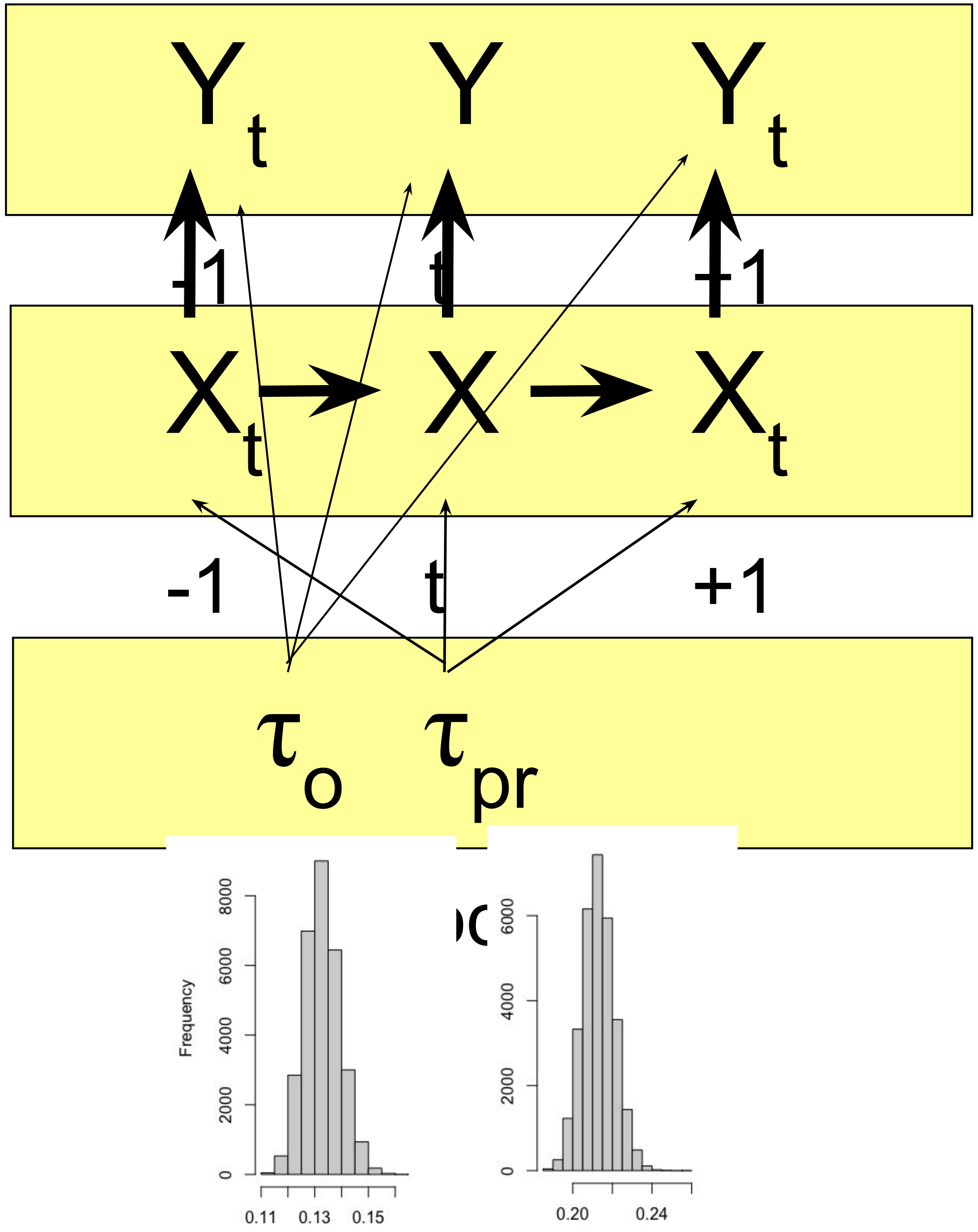
count	time
NA	4/2000
400	5/2000

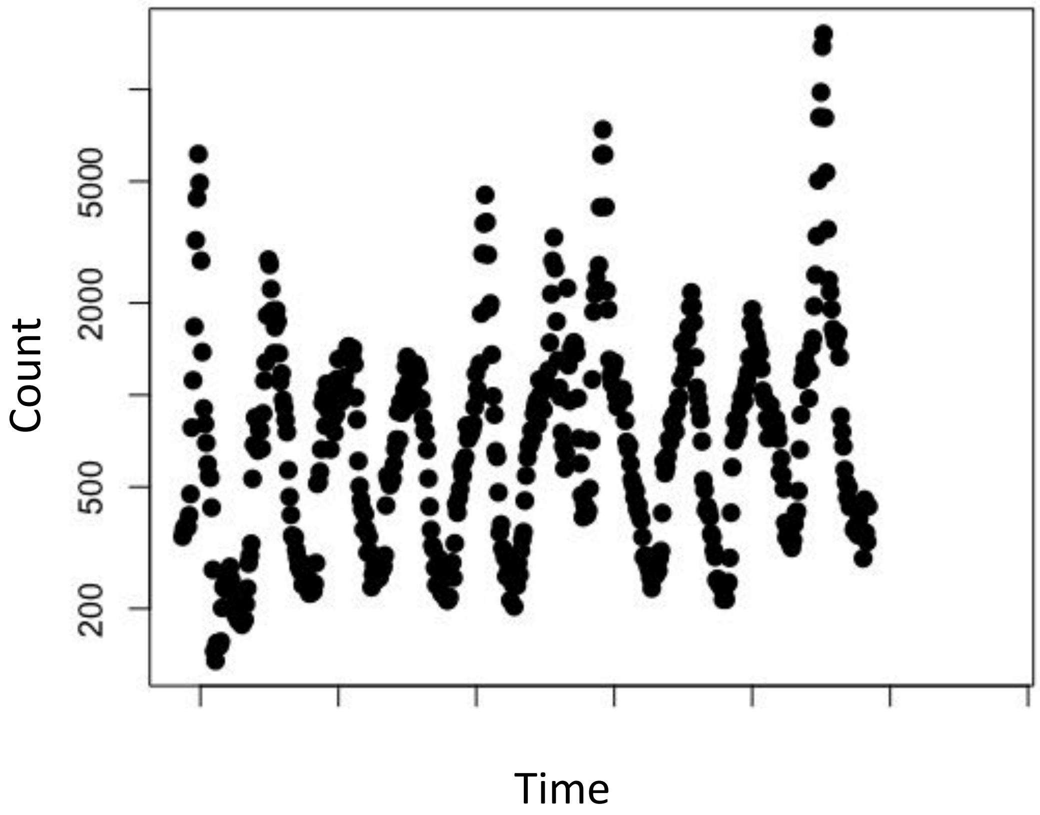


count	time
NA	4/2000
400	5/2000

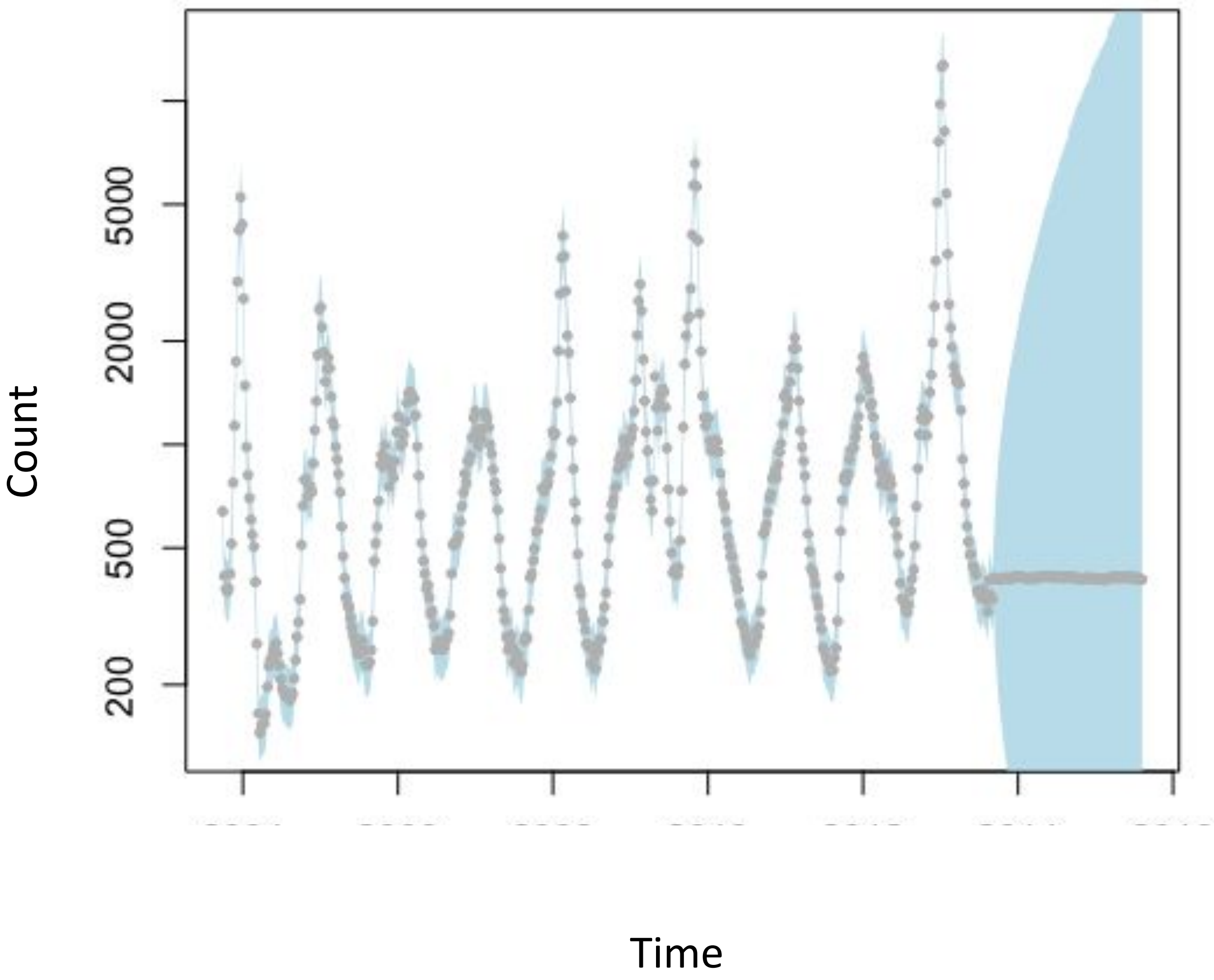
```
RandomWalk = "
model{

  #### Data Model
  for(t in 1:n){
    y[t] ~ dnorm(x[t],tau_obs)
  }

  #### Process Model
  for(t in 2:n){
    x[t]~dnorm(x[t-1],tau_proc)
  }

  #### Priors
  x[1] ~ dnorm(x_ic,tau_ic)
  tau_obs ~ dgamma(a_obs,r_obs)
  tau_proc ~ dgamma(a_proc,r_proc)
}
```


Count
5000
2000
500
200

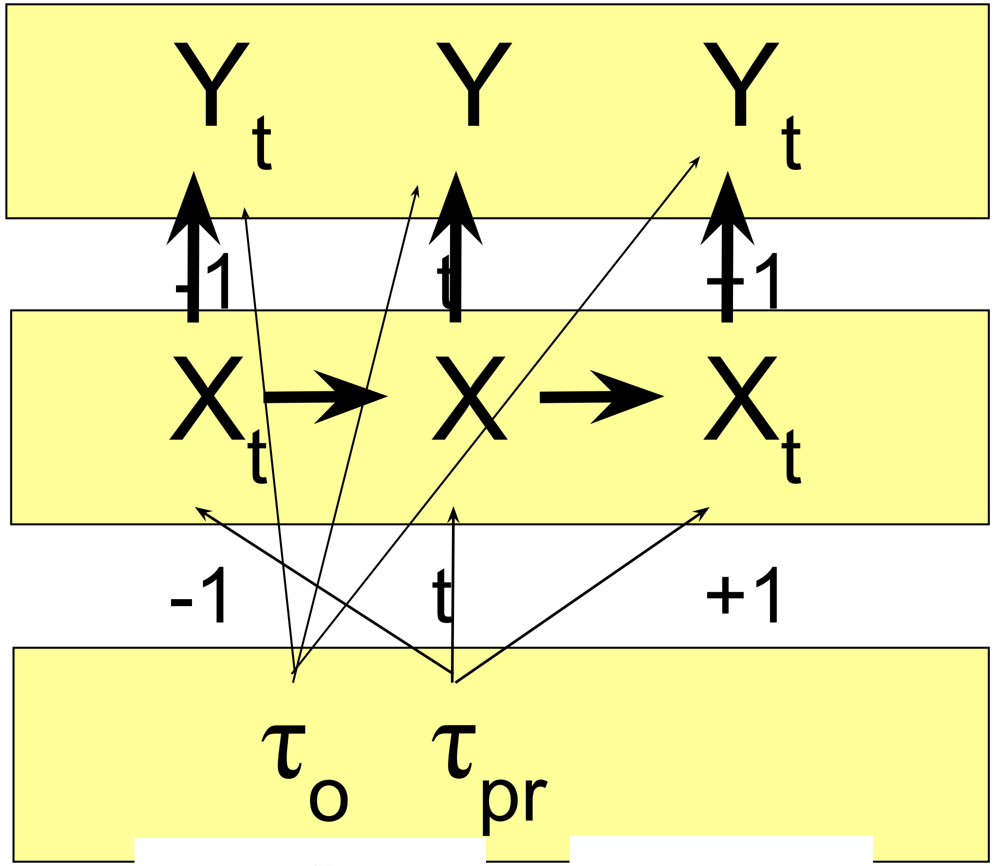

Y
t
Y
Y
t
-1
t
+1
X
t
X
X
t
-1
t
+1
τ_o
τ_{pr}
Frequency
0
2000
4000
6000
8000
0.11
0.13
0.15
0
2000
4000
6000
0.20
0.24



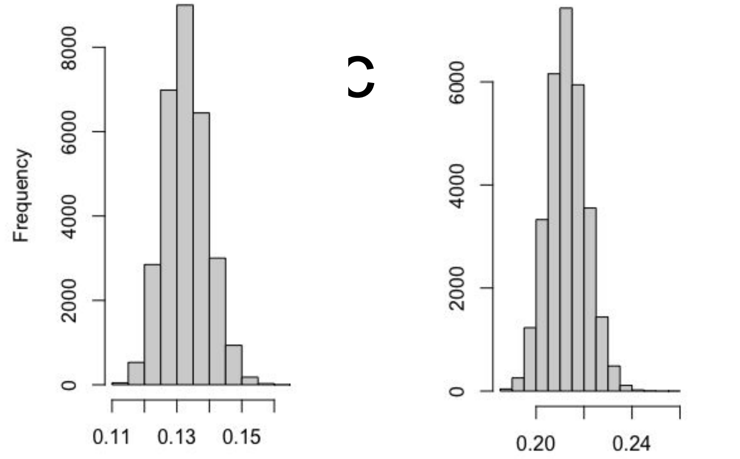
Count
5000
2000
500
200
Time



Count
5000
2000
500
200
Time



Y_t
Y
Y_t
X_t
X
X_t
-1
t
+1
τ_o
τ_{pr}



Frequency
8000
6000
4000
2000
0
0.11
0.13
0.15
6000
4000
2000
0
0.20
0.24

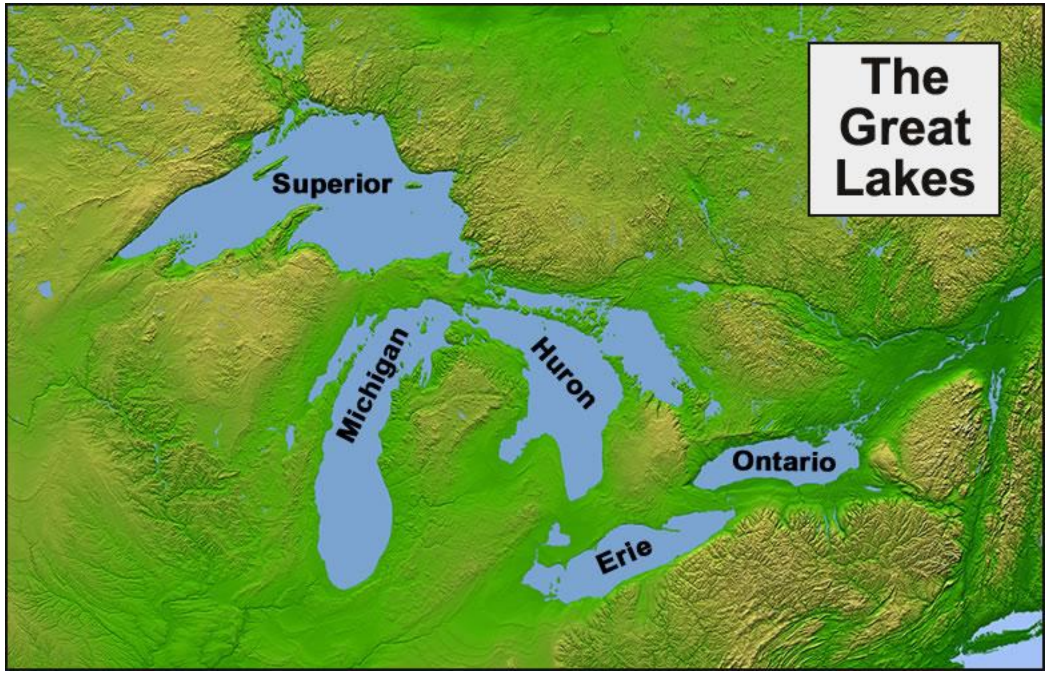
State Space Model

- Easily handles missing data (gaps)
- Neither X nor Y need be Normal

How would you make this a Poisson Random Walk ?

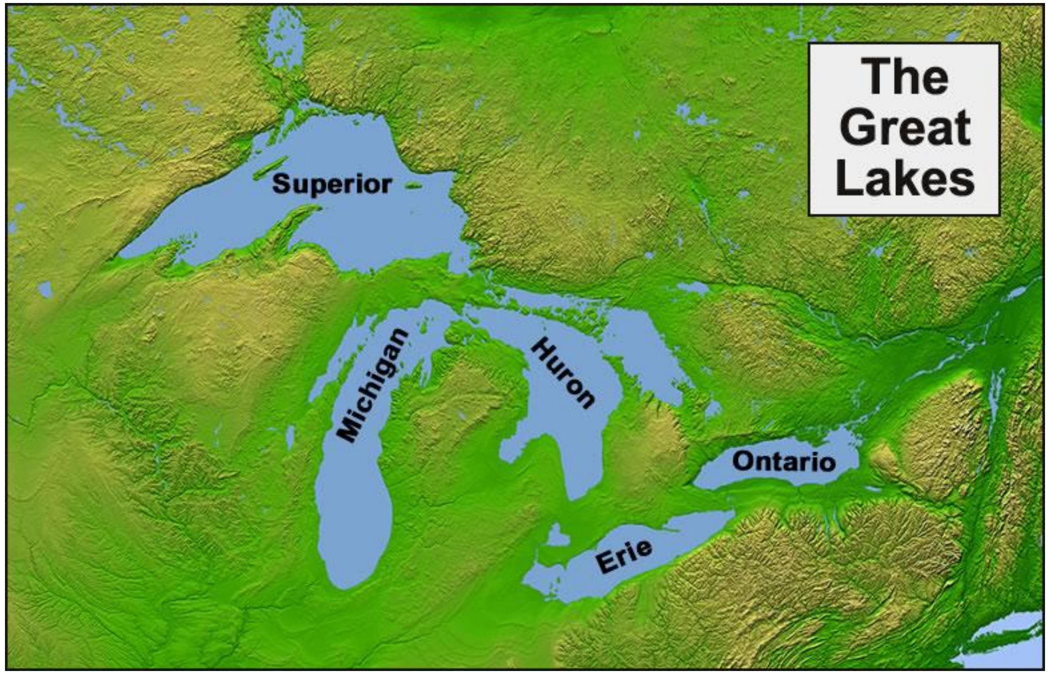
```
model{

  #### Data Model
  for(t in 1:n){
    y[t] ~ dnorm(x[t],tau_obs)
  }

  #### Process Model
  for(t in 2:n){
    x[t]~dnorm(x[t-1],tau_proc)
  }

  #### Priors
  x[1] ~ dnorm(x_ic,tau_ic)
  tau_obs ~ dgamma(a_obs,r_obs)
  tau_proc ~ dgamma(a_proc,r_proc)
}
```

Y = time series (ndates) of algal counts density each lake (sites)



```
RandEffs = "
model{
 #### Data Model
for(t in 1:ndates){
   y[t] ~ dnorm(x[t],tau_obs)
 }
 #### Process Model
 for(t in 2:ndates){
   x[t]~dnorm(mu[t],tau_add)
   mu[t]<- x[t-1] + alpha.sp[site[t]]
 }
#### Priors
x[1] ~ dnorm(ic1,ic2)##initial condition
}
     tau_add ~ dgamma(ta1,ta2)
     tau_obs ~ dgamma(to1,to2)
 sigma2<-1/tau_obs

for(j in 1:sites){
 alpha.sp[j]~dnorm(0,tau_alpha.sp)
}
tau_alpha.sp ~ dgamma(1.5,1E-4)  #weight zero

 }
```

Y = time series (ndates) of algal density & water temps for each lake (sites)

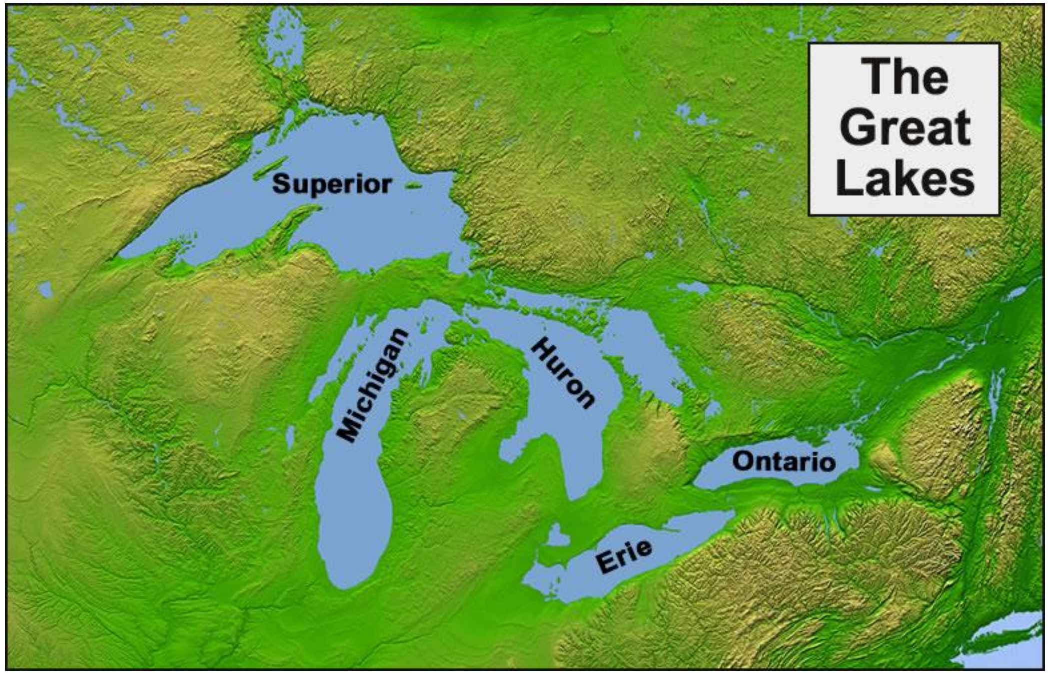


```
RandEffs = "
model{
 #### Data Model
for(t in 1:ndates){
   y[t] ~ dnorm(x[t],tau_obs)
 }
 #### Process Model
 for(t in 2:ndates){
   x[t]~dnorm(mu[t],tau_add)
   mu[t]<- b + b[1]*x[t-1] + b[2]+temp[t] + alpha.sp[site[t]]
 }
for(j in 1:sites){
 alpha.sp[j]~dnorm(0,tau_alpha.sp)
}
#### Priors
x[1] ~ dnorm(ic1,ic2)##initial condition
}
```

MISSING?

```
     tau_add ~ dgamma(ta1,ta2)
     tau_obs ~ dgamma(to1,to2)
     tau_alpha.sp ~ dgamma(1.5,1E-4)  #weight zero

 }
```

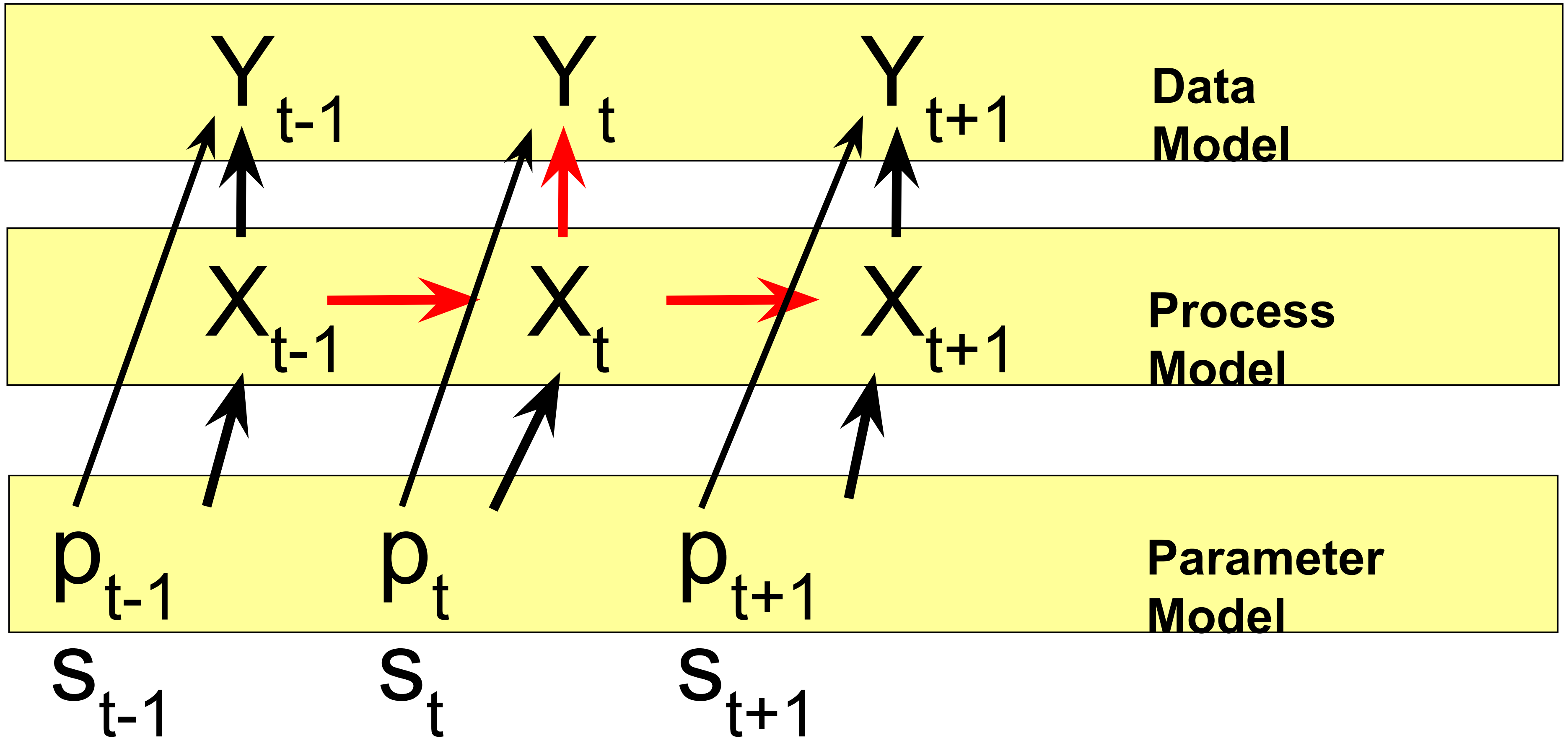
Y = time series (ndates) of algal density & water temps for each lake (sites)



Capture-Recapture

- Individuals captured, marked, and released with goal of estimating population size.
- Over repeated censuses will recapture some fraction of the population
- Assume recapture is random

Mark Recapture State Space



Capture-Recapture

- Suppose an individual record consists of capture data

$$Y_i = [1,0,1,0,0]$$

- This is compatible with the following survival

$$X_i = [1,1,1,0,0]$$
$$X_i = [1,1,1,1,0]$$
$$X_i = [1,1,1,1,1]$$

Basic Mark-Recapture State Space

- Process model

$$P(X_t = 1 \mid X_{t-1} = 1) = s_t$$

$$P(X_t = 1 \mid X_{t-1} = 0) = 0$$

$$P(X_t = 0 \mid X_{t-1} = 1) = 1 - s_t$$

$$P(X_t = 0 \mid X_{t-1} = 0) = 1$$

Bernoulli Survival Probability

Basic Mark-Recapture State Space

- Process model

$$P(X_t = 1 \mid X_{t-1} = 1) = s_t$$
$$P(X_t = 1 \mid X_{t-1} = 0) = 0$$
$$P(X_t = 0 \mid X_{t-1} = 1) = 1 - s_t$$
$$P(X_t = 0 \mid X_{t-1} = 0) = 1$$

Bernoulli Survival Probability

$s^x (1-s)^{1-x}$

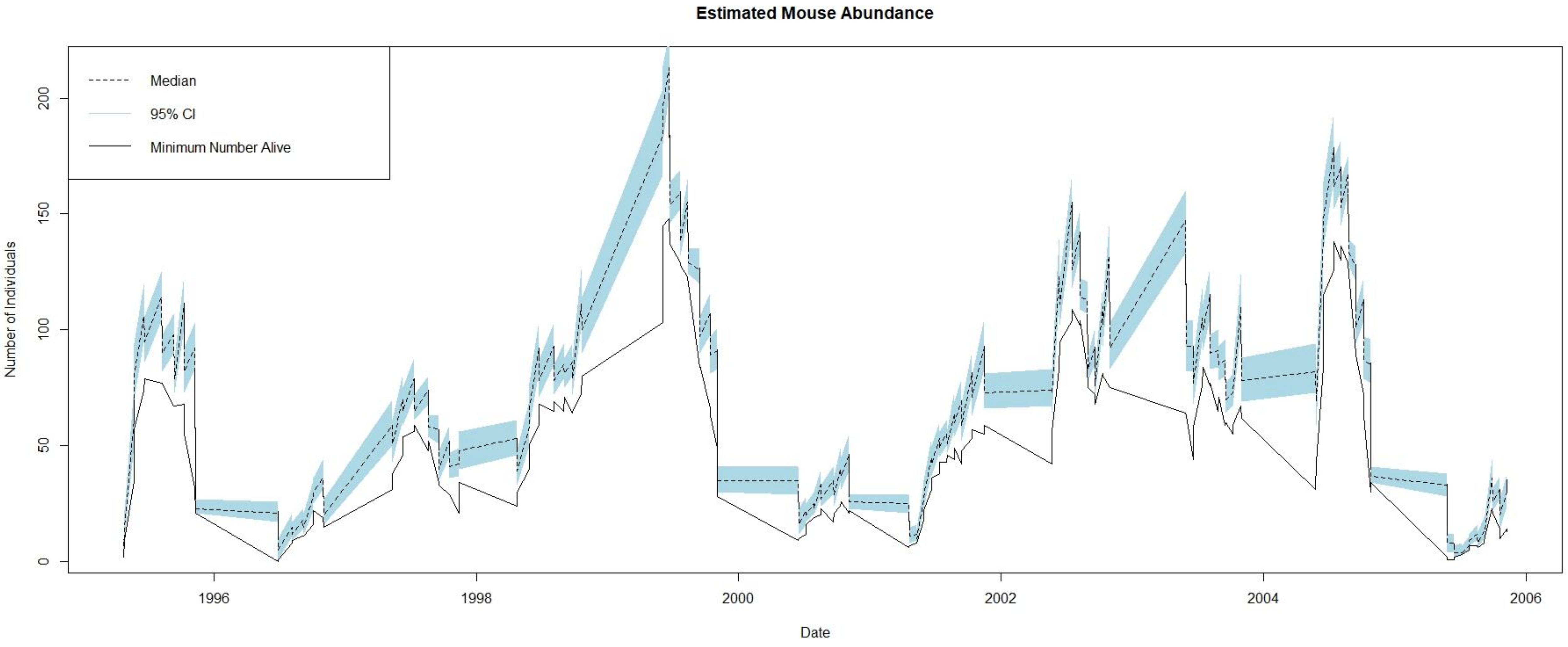
- Observation model

$$P(Y_t = 1 \mid X_t = 1) = p_t$$
$$P(Y_t = 1 \mid X_t = 0) = 0$$
$$P(Y_t = 0 \mid X_t = 0) = 1$$
$$P(Y_t = 0 \mid X_t = 1) = 1 - p_t$$

Bernoulli Detection Probability

$p^x (1-p)^{1-x}$

- Priors on p and s (e.g. Beta)



Estimated Mouse Abundance
Median
95% CI
Minimum Number Alive
Number of Individuals
0
50
100
150
200
1996
1998
2000
2002
2004
2006
Date

John Foster