Temporal autocorrelation

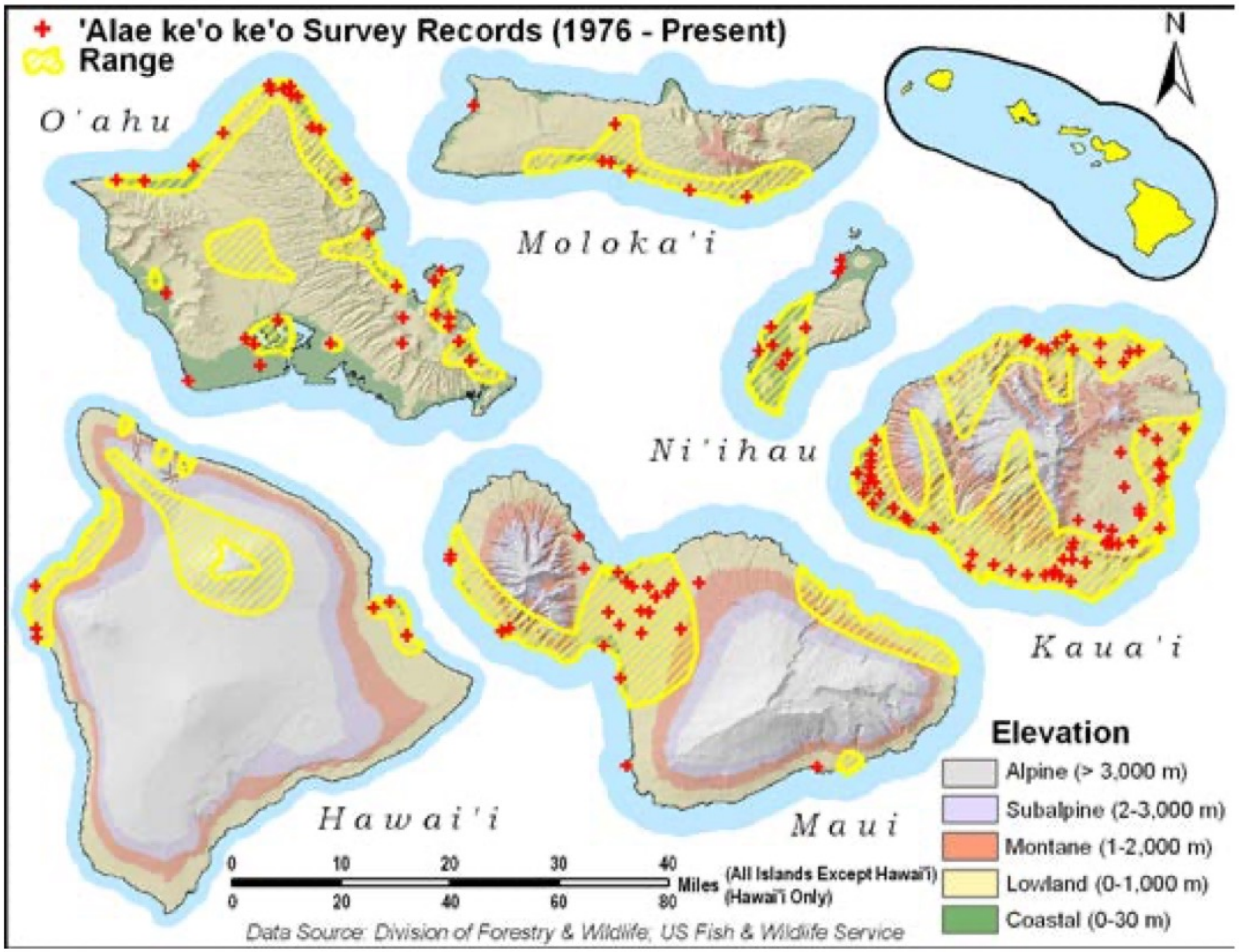
Lots of ways to analyze/model time series data:

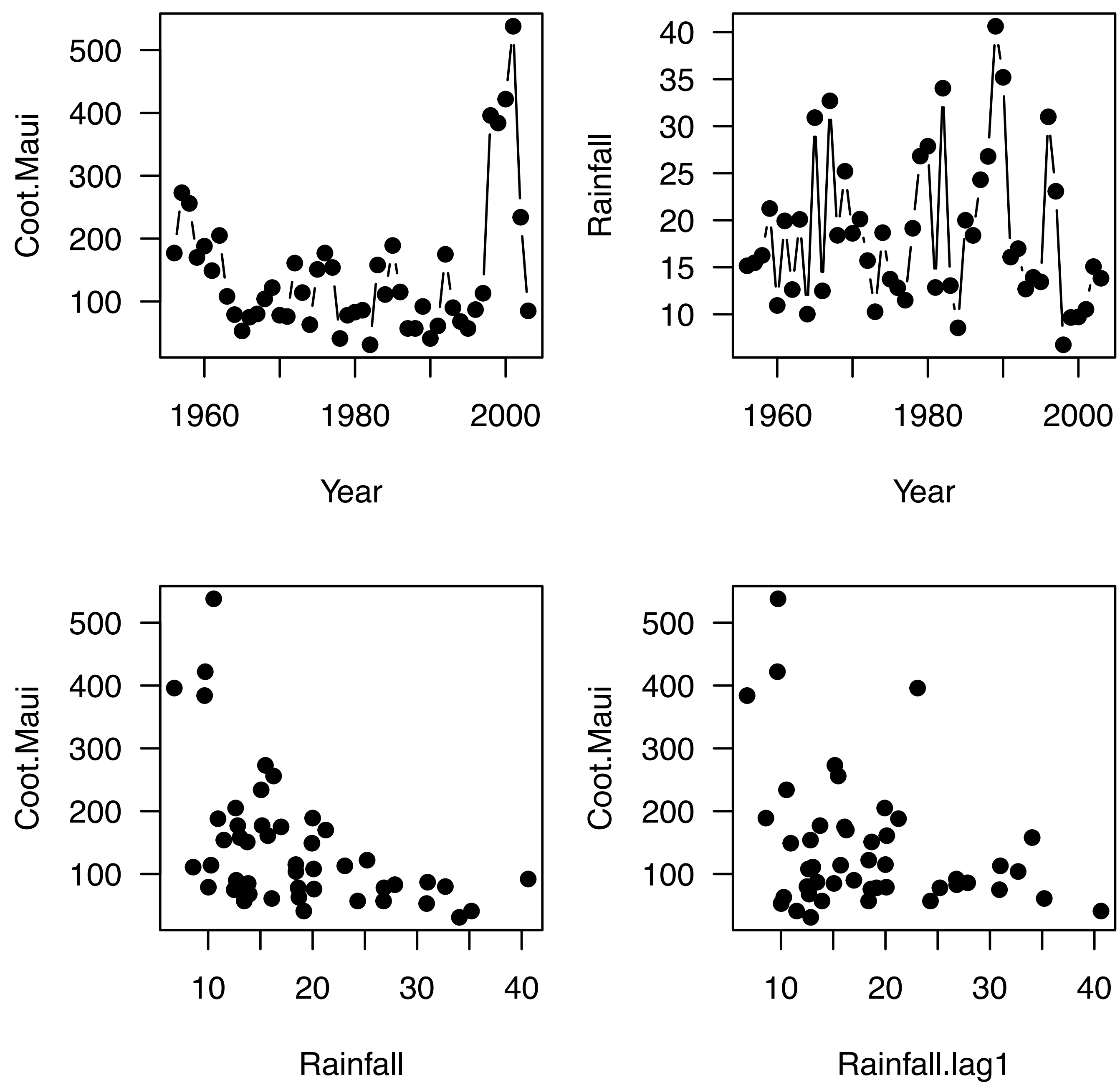
- Autoregressive
- Spectral analysis
- Mechanistic population dynamics

How to use time series data in LM, GLM, GLMM, etc?

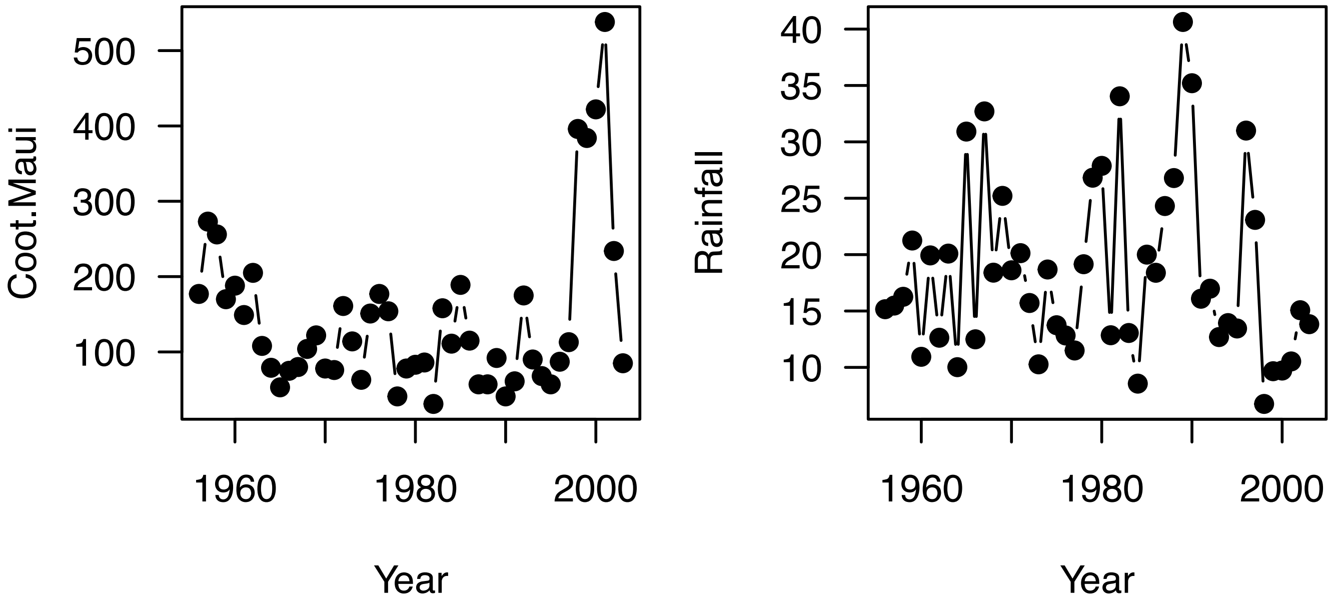
‘alae ke‘oke‘o - Hawaiian coot - *Fulica alai*

Endemic and endangered





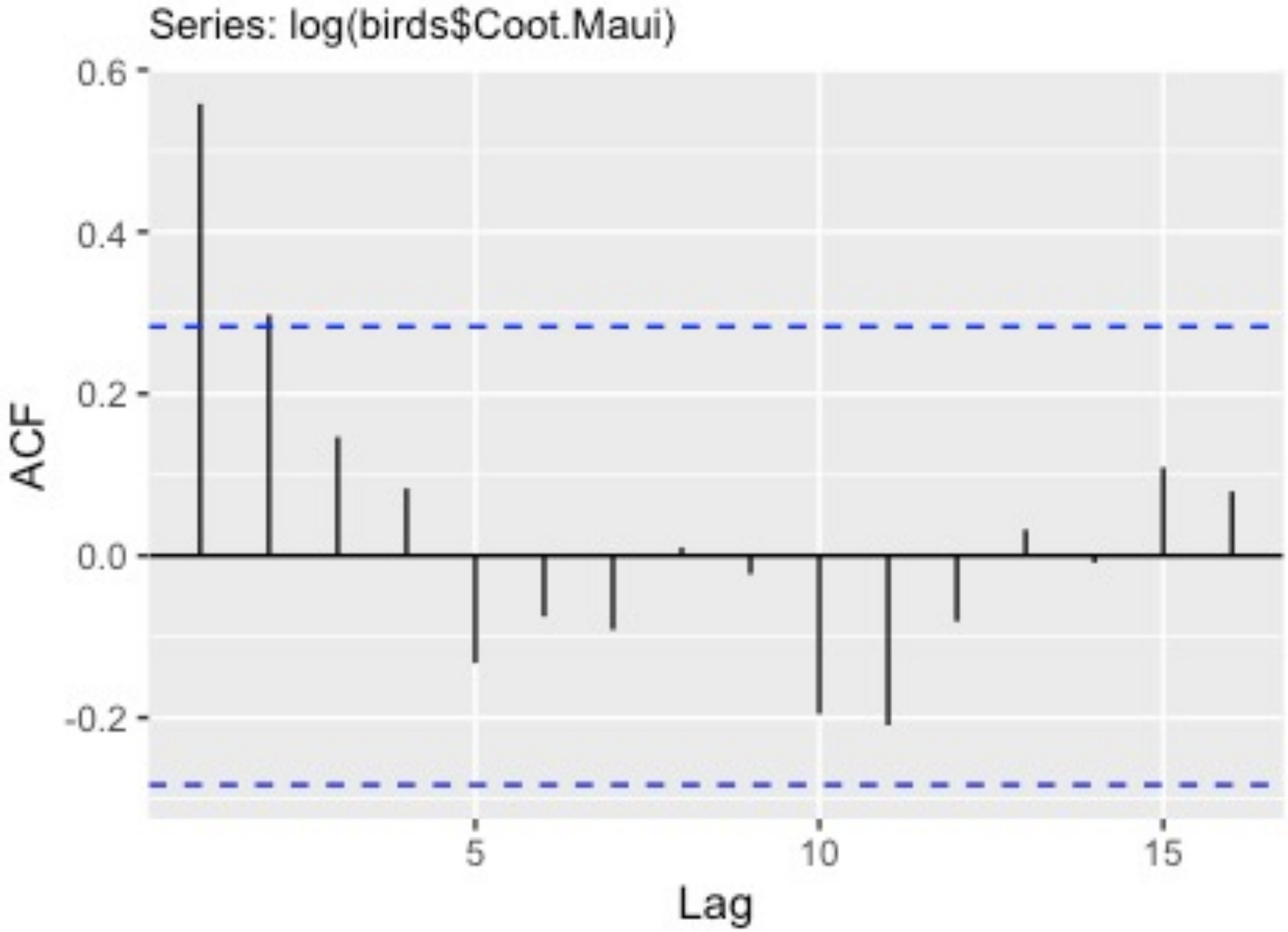
- Rainfall may affect habitat availability, but also counting/detection success
- Treat the bird counts as lognormal



Patterns in time series can be explore with autocorrelation function:

$$r_k = \frac{\sum_{t=1}^{N-k} (Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{\sum_{t=1}^{N} (Y_t - \bar{Y})^2}$$

```
ggAcf(log(birds$Coot.Maui))
```



Why might abundance be correlated at 1-2 year lag?

How can we model this kind of dependence, using only the time series itself?

Autoregressive model

The observation at time *t* depends on one or more previous observations

$$Y_t = c + \varphi Y_{t-1} + \varepsilon_t$$

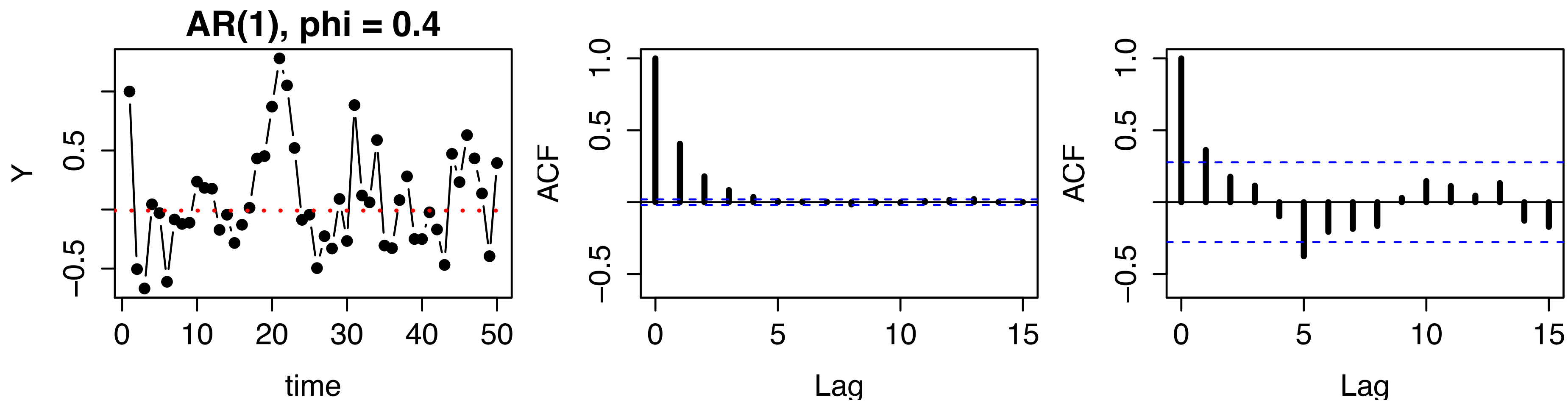
Usually assume phi is between -1 and 1; represents the lag-1 autocorrelation

This is **AR(1)**

AR(2):

$$Y_t = c + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \varepsilon_t$$

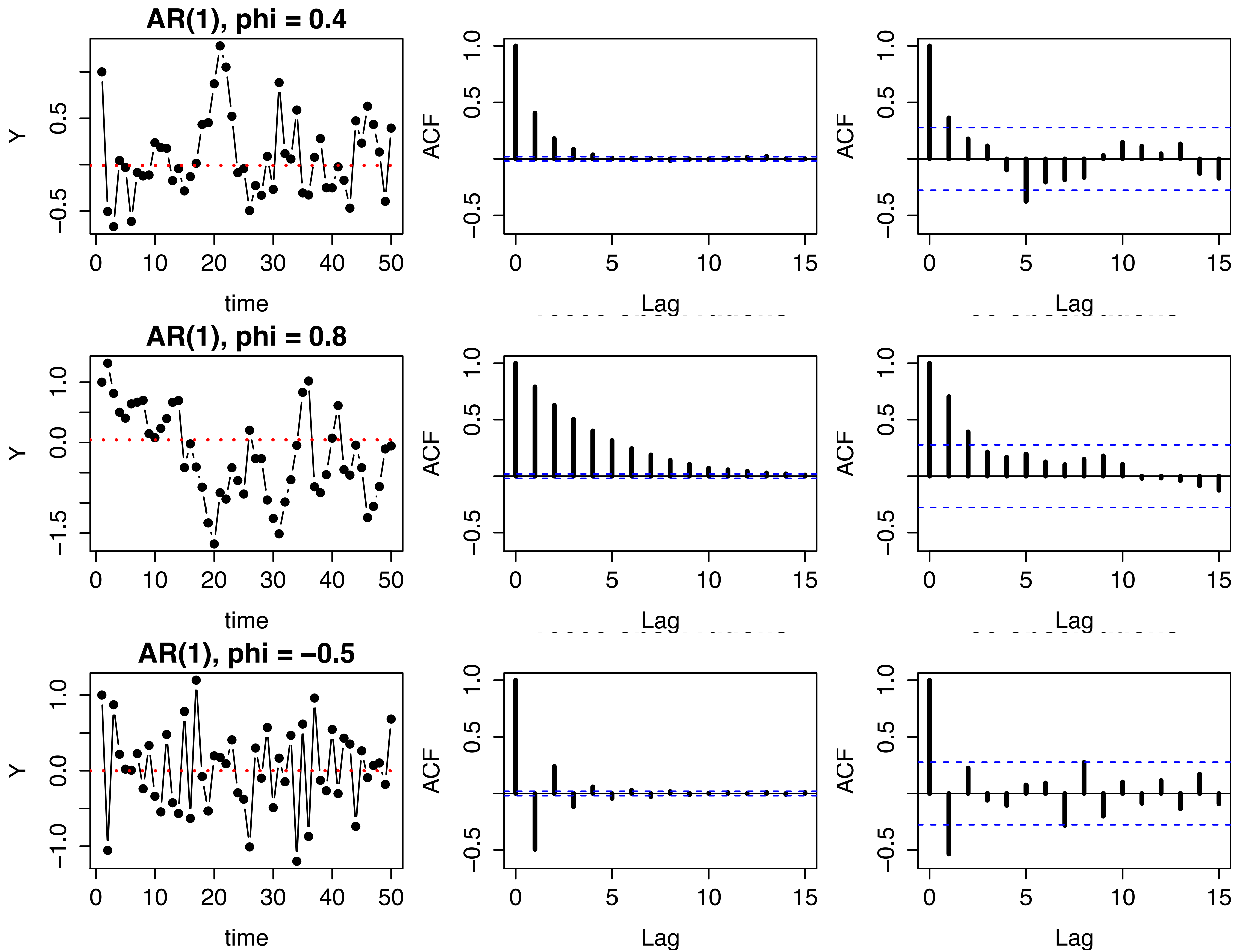
```r
#AR1
phis = c(0.4, 0.8, -0.5)
par(mfrow = c(3,3), mar = c(4,4,2,1))
for (j in 1:3) {
Y = 1
tmax = 10000
SD = 0.5
phi = phis[j]
for (t in 2:tmax) {
  Y[t] = phi*Y[t-1] + rnorm(1, mean = 0, sd = SD)
}

plot(c(1:50), Y[1:50], type = 'b', pch = 19, main = paste('AR(1), phi =', phi)
 xlab = 'time', ylab = 'Y')
abline(h = mean(Y), col = 'red', lty = 3, lwd = 2)
acf(Y, lwd = 3, lag.max = 15, ylim = c(-0.3, 1), main = '10000 observations')
acf(Y[1:50], lwd = 3, lag.max = 15, ylim = c(-0.3, 1), main = '50 observations
')
}
```



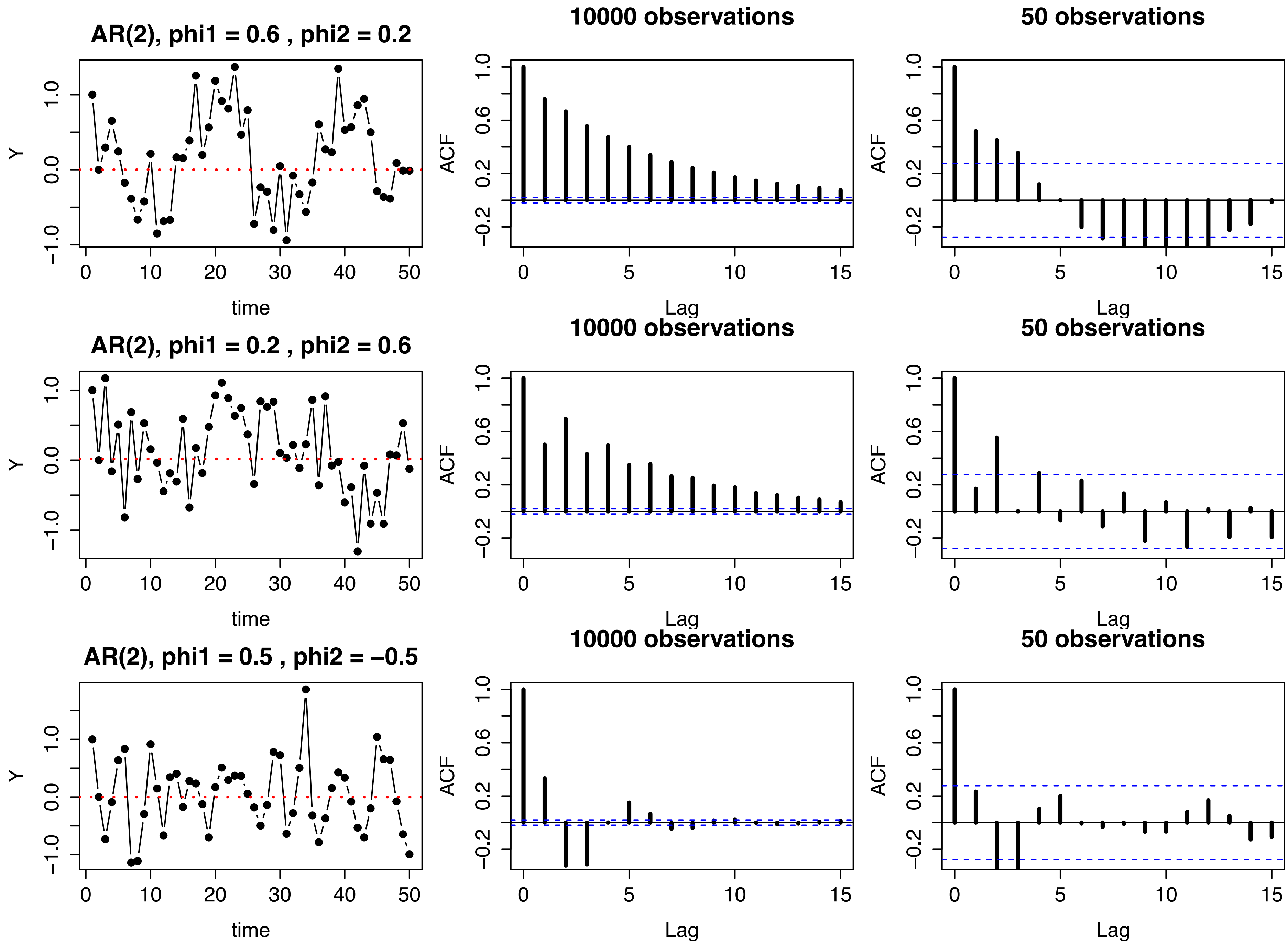
$$Y_2 = \varphi Y_1 + \varepsilon_2$$
$$Y_3 = \varphi Y_2 + \varepsilon_3 = \varphi(\varphi Y_1 + \varepsilon_2) + \varepsilon_3 = \varphi^2 Y_1 + \varphi\varepsilon_2 + \varepsilon_3$$

Lag-1	0.4
Lag-2	$0.4^2 = 0.16$
Lag-3	$0.4^3 = 0.064$

E.g., a species that lives for a few years



AR(1), phi = 0.4
AR(1), phi = 0.8
AR(1), phi = −0.5
Y
time
ACF
Lag

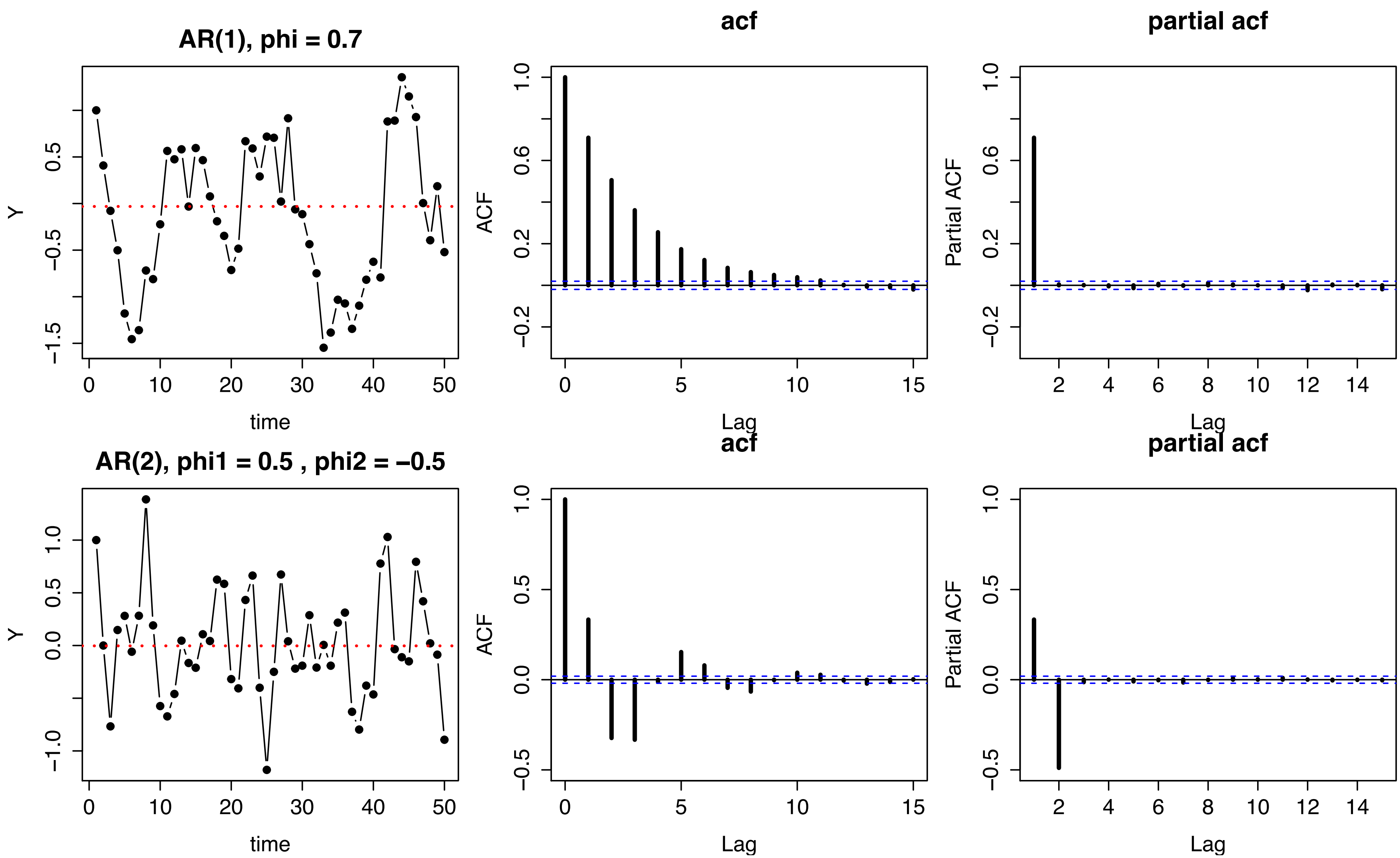


AR(2), phi1 = 0.6 , phi2 = 0.2
10000 observations
50 observations
AR(2), phi1 = 0.2 , phi2 = 0.6
10000 observations
50 observations
AR(2), phi1 = 0.5 , phi2 = −0.5
10000 observations
50 observations
Y
time
ACF
Lag

Partial autocorrelation function:

Correlation at lag *k*, after removing the effect of earlier lags

acf(mydata, type = “partial”)



Moving average model

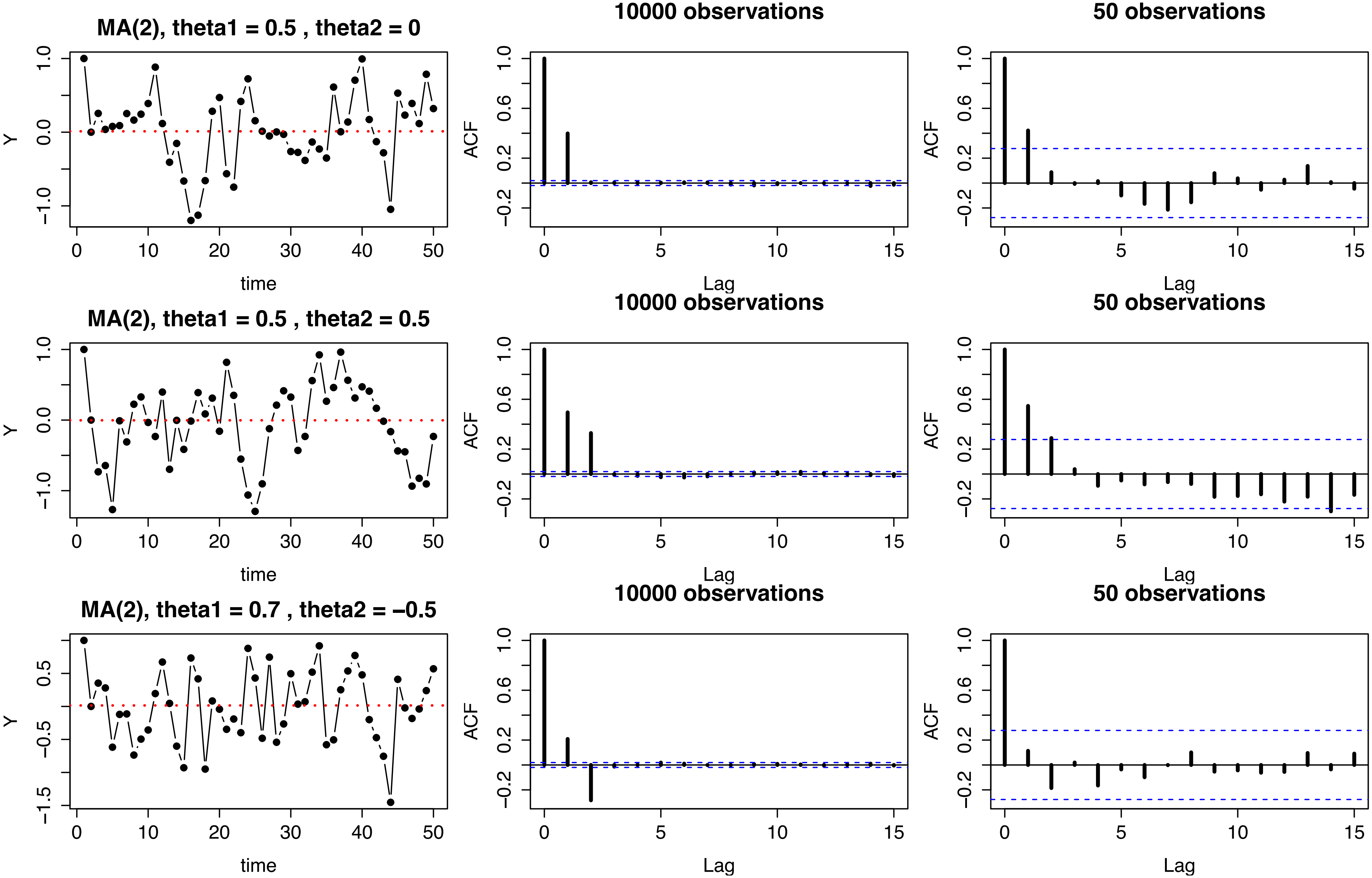
Value at time *t* doesn't depend directly on previous values

Depends on random 'shocks' from previous timesteps

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

Connection to population dynamics less obvious, but good at modeling short-term autocorrelation that does not have a 'memory'

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$



In principle, can distinguish AR and MA, and their order, using acf

But with limited data it’s not so clear. Try simple models, compare, add compexity.

Generalized least squares (GLS)

We want to 1) model the effect of rainfall on abundance, 2) while also accounting for temporal autocorrelation

We will do this by adding a **model for the residuals**

Terminology is unfortunate:

- GLS is for normal data with correlated residuals

- GLM for non-normal data with uncorrelated residuals

- GEE = Generalized Estimating Equations is for non-normal data with correlated resids

$$Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$$

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \end{bmatrix} = \vec{\varepsilon} \sim MultivariateNormal(\Sigma)$$

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \varphi & \varphi^2 & \varphi^3 \\ \varphi & 1 & \varphi & \varphi^2 \\ \varphi^2 & \varphi & 1 & \varphi \\ \varphi^3 & \varphi^2 & \varphi & 1 \end{bmatrix}$$

Sigma² is the residual variance

Phi is the correlation between consecutive residuals

This is an **AR(1)** model

Note: we assume the amount of variability does not change over time

Stationary: the distribution from which the data are drawn does not change over time

```
mod.nopredictors = gls(log(Coot.Maui) ~ 1, data = birds, correlation =
corAR1(form =~ Year))
```

Syntax like lm(), but with a correlation argument

See ?corStruct

AR1 has its own function

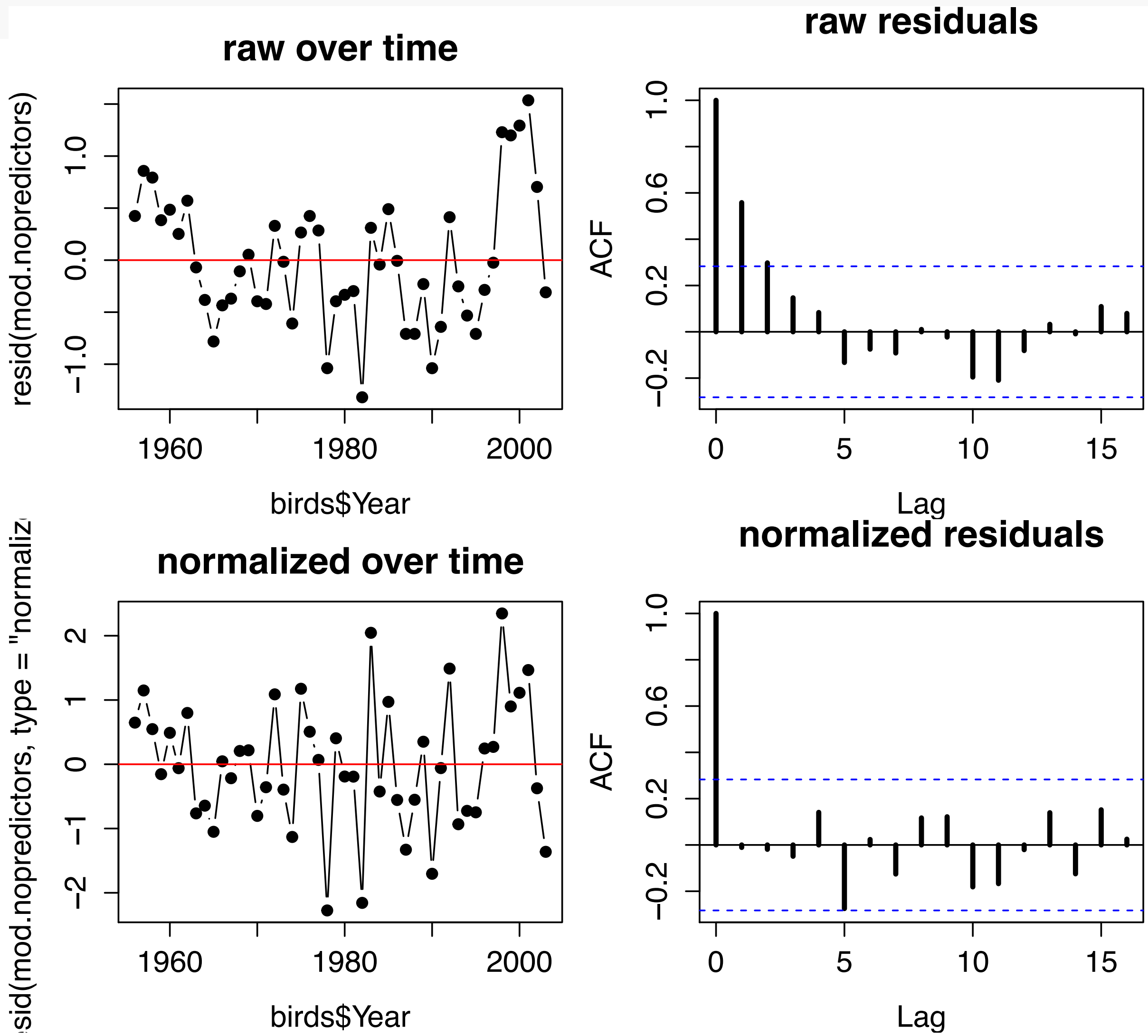
Can also do AR1 within groups: form =~ Year | Site

```
## Generalized least squares fit by REML
##   Model: log(Coot.Maui) ~ 1
##   Data: birds
##    AIC   BIC logLik
##   82.3 87.85 -38.15
##
## Correlation Structure: AR(1)
##  Formula: ~Year
##  Parameter estimate(s):
##   Phi
## 0.587
##
## Coefficients:
##             Value Std.Error t-value p-value
## (Intercept) 4.752      0.18    26.4       0
##
## Standardized residuals:
##      Min       Q1      Med       Q3      Max
## -2.01305 -0.61361 -0.08572  0.63491  2.34595
##
## Residual standard error: 0.6547
## Degrees of freedom: 48 total; 47 residual
```

Can compare raw residuals and ‘normalized’ residuals

$$Y_t - \text{fitted}(Y_t) = \eta_t = \varphi\eta_{t-1} + \varepsilon_t$$

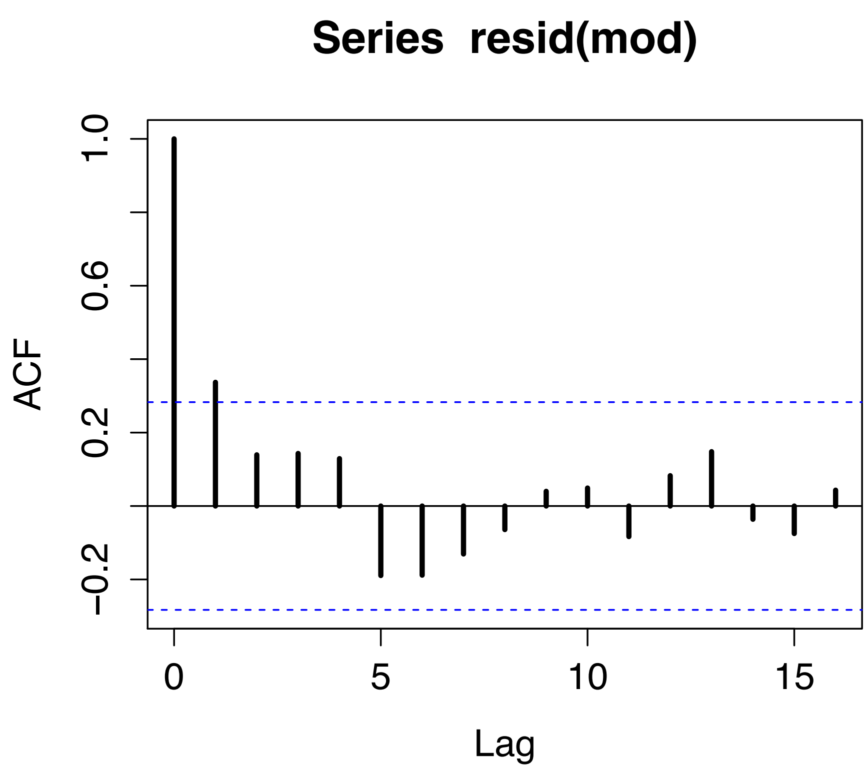
```r
plot(resid(mod.nopredictors) ~ birds$Year, type = 'b', pch = 19, main = 'raw o
ver time')
abline(h = 0, col = 'red')
plot(resid(mod.nopredictors, type = "normalized") ~ birds$Year, type = 'b', pc
h = 19, main = 'normalized over time')
abline(h = 0, col = 'red')
acf(resid(mod.nopredictors), lwd = 3, main = 'raw residuals')
acf(resid(mod.nopredictors, type = "normalized"), lwd = 3, main = 'normalized
residuals')
```



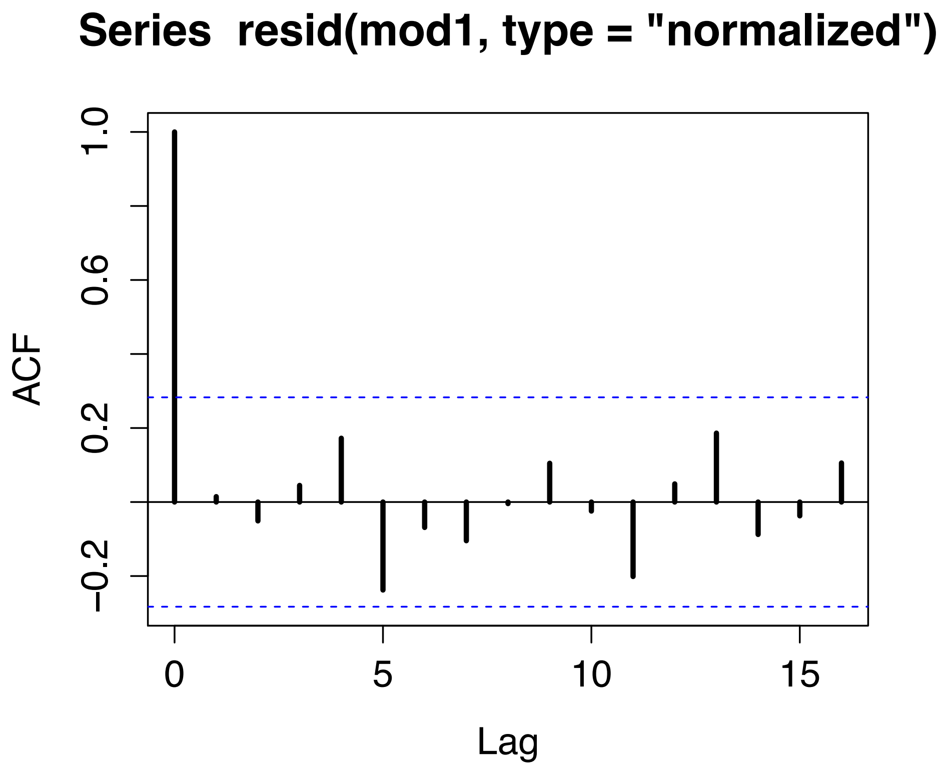
Important: model assumes residuals independent after accounting for the predictors

So using the right predictor may be sufficient (like we saw with the spatial smoother)≈

```
mod = lm(log(Coot.Maui) ~ Rainfall, data = birds)
acf(resid(mod))
```

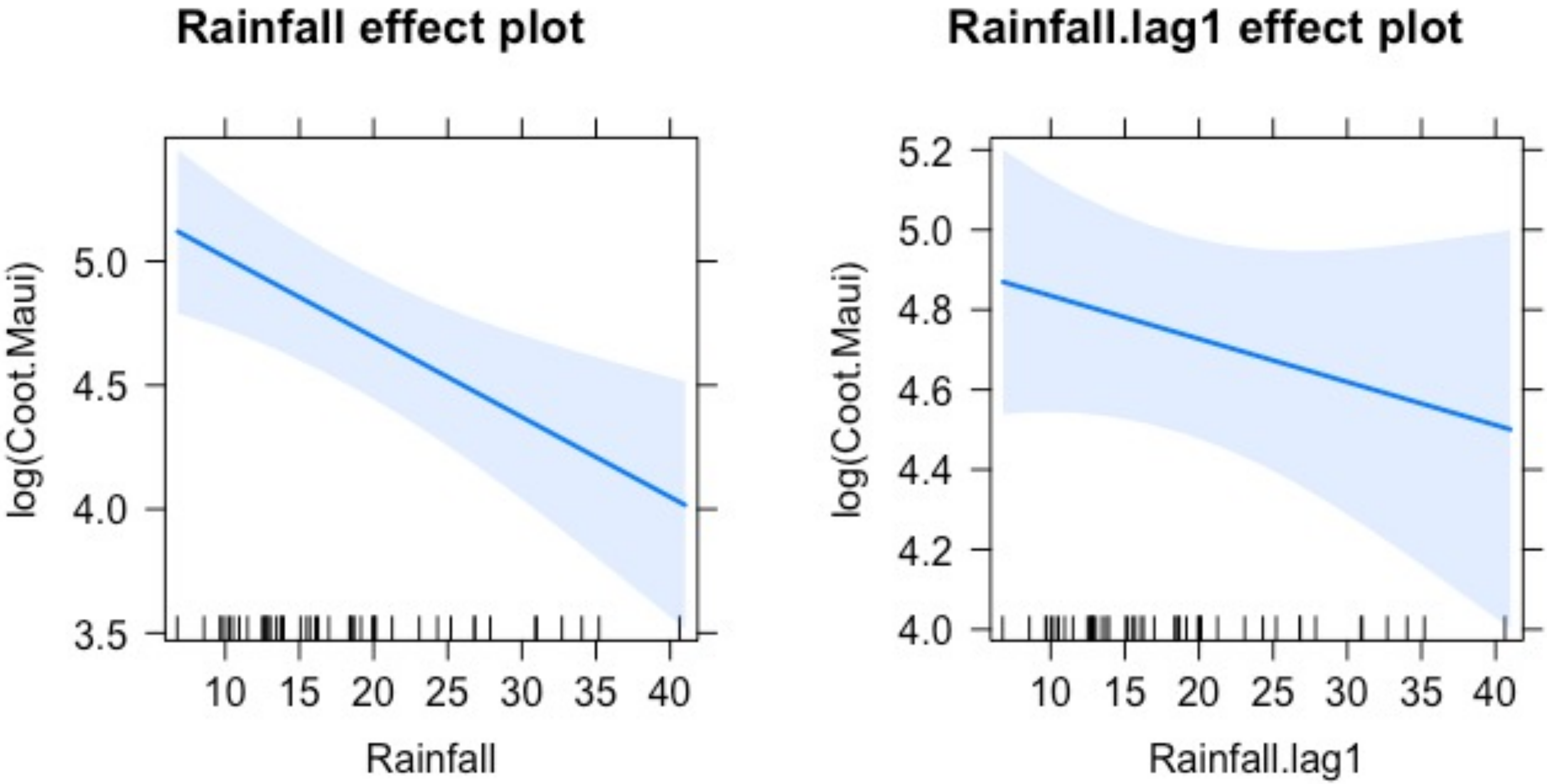


```
mod1 = gls(log(Coot.Maui) ~ Rainfall, data = birds, correlation = corAR1(form
=~ Year))
acf(resid(mod1, type = "normalized"))
```



```
  Model: log(Coot.Maui) ~ Rainfall + Rainfall.lag1
  Data: subset(birds, !is.na(Rainfall.lag1))
       AIC      BIC    logLik
  74.40192 83.65266 -32.20096

Correlation Structure: AR(1)
 Formula: ~Year
 Parameter estimate(s):
      Phi
0.4161663

Coefficients:
                  Value  Std.Error   t-value p-value
(Intercept)    5.538649 0.30630520 18.082126  0.0000
Rainfall      -0.032254 0.00949971 -3.395276  0.0015
Rainfall.lag1 -0.010790 0.00950933 -1.134696  0.2626

 Correlation:
              (Intr) Ranfll
Rainfall      -0.737
Rainfall.lag1 -0.739  0.294

Standardized residuals:
        Min          Q1         Med          Q3         Max
-2.05602659 -0.81469555  0.06235979  0.62591168  2.26555967

Residual standard error: 0.5269013
Degrees of freedom: 47 total; 44 residual
```

```
> anova(mod1)
Denom. DF: 44
               numDF   F-value p-value
(Intercept)        1 1517.5655  <.0001
Rainfall           1   10.2611  0.0025
Rainfall.lag1      1    1.2875  0.2626
```



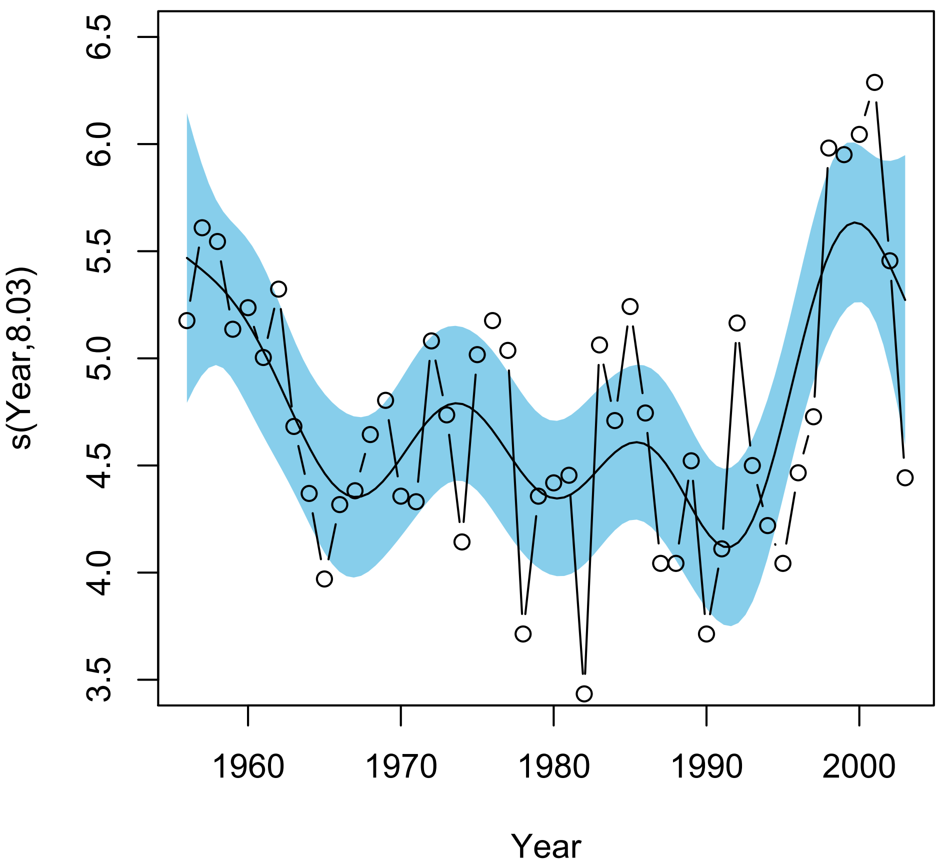
exp(5) = 148 to exp(4) = 55

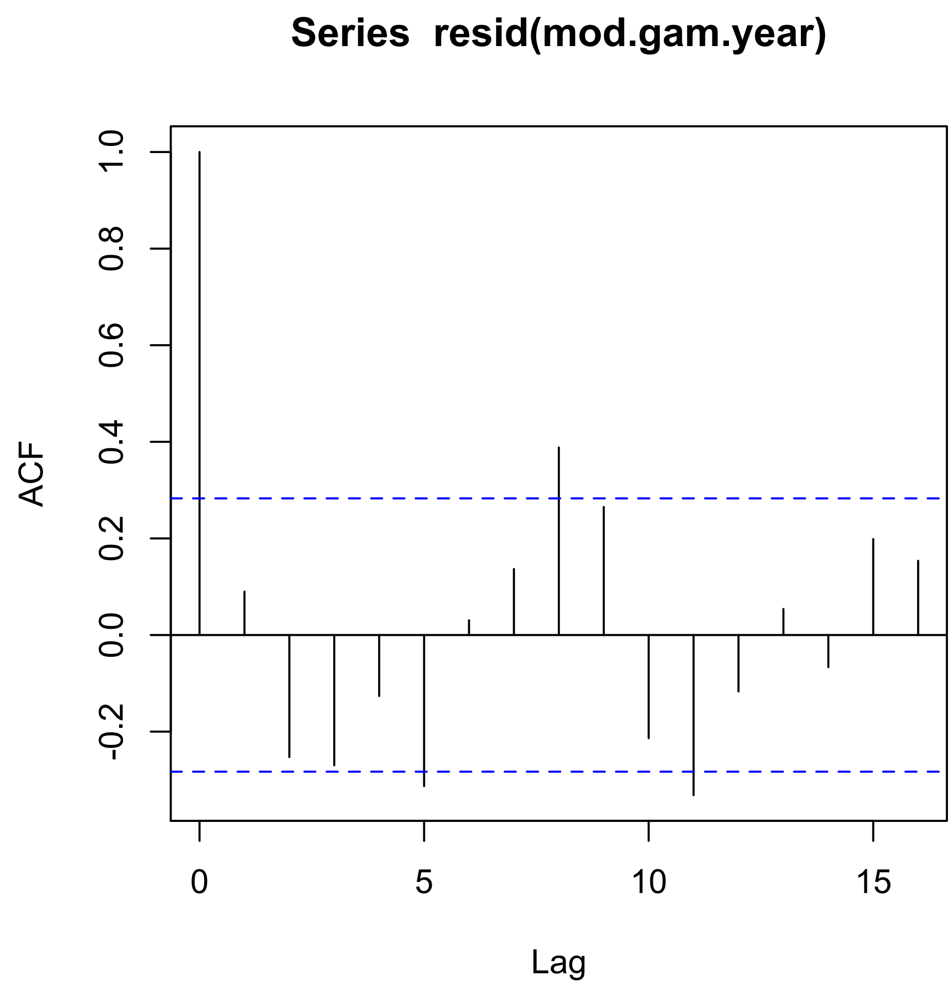
Conclusions?

- High rainfall in current year may make detection harder (more habitat)
- High rainfall in previous year doesn't seem to translate into a larger population this year

Alternative to GLS: can we account for temporal autocorrelation using a smoother?

```
mod.gam.year = gam(log(Coot.Maui) ~ s(Year), data = birds)
```



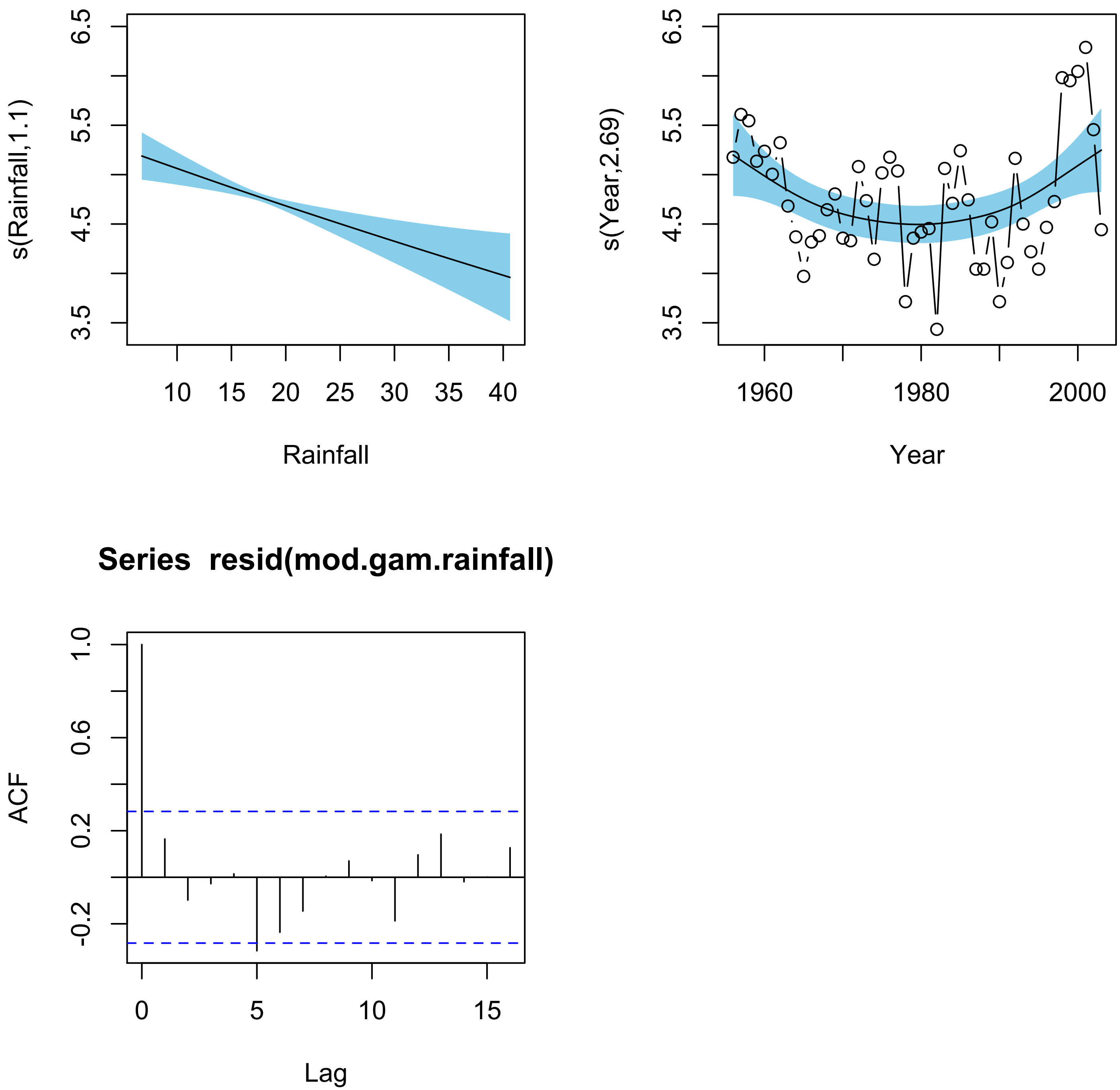


Alternative to GLS: can we account for temporal autocorrelation using a smoother?

```
mod.gam.year = gam(log(Coot.Maui) ~ s(Year), data = birds)
```

```
> gam.check(mod.gam.year)

Method: GCV   Optimizer: magic
Smoothing parameter selection converged after 9 iterations.
The RMS GCV score gradiant at convergence was 4.240182e-07 .
The Hessian was positive definite.
The estimated model rank was 10 (maximum possible: 10)
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

           k'   edf k-index p-value
s(Year) 9.000 8.033   0.886    0.14
```

Alternative to GLS: can we account for temporal autocorrelation using a smoother?



Alternative to GLS: can we account for temporal autocorrelation using a smoother?

```
mod.gam.rainfall = gam(log(Coot.Maui) ~ s(Rainfall) + s(Year), data = birds)
```

```
Family: gaussian
Link function: identity

Formula:
log(Coot.Maui) ~ s(Rainfall) + s(Year)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.7485     0.0694   68.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
              edf Ref.df      F  p-value
s(Rainfall) 1.102  1.197 12.200 0.000559 ***
s(Year)     2.686  3.349  3.544 0.018407 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.437   Deviance explained = 48.2%
GCV = 0.25677  Scale est. = 0.23116   n = 48
```

For non-normal data: Generalized Estimating Equations

Basically an extension of quasipoisson, quasibinomial

For Poisson, assume the variance is proportional to the mean, modified by the residual correlation structure you specify

```
library(geepack)

birds$idvar = factor(rep('1', nrow(birds)))

gee.rainfall = geeglm(Coot.Maui ~ Rainfall, family = poisson, id = idvar, corstr = 'ar1',
data = birds)
```

For non-normal data: Generalized Estimating Equations

Basically an extension of quasipoisson, quasibinomial

For Poisson, assume the variance is proportional to the mean, modified by the residual correlation structure you specify

```
Call:
geeglm(formula = Coot.Maui ~ Rainfall, family = poisson, data = birds,
    id = idvar, corstr = "ar1")

 Coefficients:
             Estimate   Std.err     Wald Pr(>|W|)
(Intercept)  5.45e+00  3.95e-07 1.90e+14   <2e-16 ***
Rainfall    -2.95e-02  1.77e-08 2.78e+12   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Estimated Scale Parameters:
            Estimate  Std.err
(Intercept)     54.9 1.27e-06

Correlation: Structure = ar1  Link = identity

Estimated Correlation Parameters:
      Estimate  Std.err
alpha    0.553 1.55e-07
Number of clusters:   1   Maximum cluster size: 48
```