Stat 565

(S)Arima & Forecasting

Feb 2 2016

Charlotte Wickham

stat565.cwick.co.nz

Today

A note from HW #3

Pick up with ARIMA processes

Introduction to forecasting

HW #3

The sample autocorrelation coefficients are biased. But asymptotically unbiased...

Theorem A.7 *If x_t is a stationary linear process of the form (1.31) satisfying the fourth moment condition (A.50), then for fixed K,*

$$\begin{pmatrix} \widehat{\rho}(1) \\ \vdots \\ \widehat{\rho}(K) \end{pmatrix} \sim AN\left[\begin{pmatrix} \rho(1) \\ \vdots \\ \rho(K) \end{pmatrix}, n^{-1}W\right],$$

where W is the matrix with elements given by

$$\begin{aligned} w_{pq} &= \sum_{u=-\infty}^{\infty} \Big[\rho(u+p)\rho(u+q) + \rho(u-p)\rho(u+q) + 2\rho(p)\rho(q)\rho^2(u) \\ &\qquad - 2\rho(p)\rho(u)\rho(u+q) - 2\rho(q)\rho(u)\rho(u+p)\Big] \\ &= \sum_{u=1}^{\infty}[\rho(u+p) + \rho(u-p) - 2\rho(p)\rho(u)] \\ &\qquad \times [\rho(u+q) + \rho(u-q) - 2\rho(q)\rho(u)], \end{aligned} \tag{A.55}$$

S&S

where the last form is more convenient.

For white noise, $W = I$,

and we have $r(h) \sim N(\rho(h), 1/n)$

Leads to CI's of the form $0 \pm 2/\sqrt{n}$ (the dashed lines in the acf plot).

HW #2 example

$$x_t = \beta_0 + \beta_1 t + w_t$$

a linear trend

$$\nabla x_t = x_t - x_{t-1} = \beta_1 + w_t - w_{t-1}$$

an MA(1) process with

constant mean β_1

x_t is called ARIMA(0, 1, 1)

ARIMA(p, d, q)

Autoregressive Integrated Moving Average

A process x_t is ARIMA(p, d, q) if x_t differenced d times ($\nabla^d x_t$) is an ARMA(p, q) process.

I.e. x_t is defined by

$$\phi(B)\, \nabla^d x_t = \theta(B)\, w_t$$

$$\phi(B)\, (1 - B)^d x_t = \theta(B)\, w_t$$

forces constant in 1st differenced series

```
arima(x, order = c(p, 1, q), xreg = 1:length(x))
```

Procedure for ARIMA modeling

We'll assume the primary goal is getting a forecast.

`diff`

1. Plot the data. Transform? Outliers? Differencing?
2. Difference until series is stationary, i.e. find d.
3. Examine differenced series and pick p and q.
4. Fit ARIMA(p, d, q) model to original data.
5. Check model diagnostics
6. Forecast (back transform?)

Pick one:

Oil prices

```
install.packages('TSA')
data(oil.price, package = 'TSA')
```

Global temperature

```
load(url("http://www.stat.pitt.edu/stoffer/tsa3/tsa3.rda"))
gtemp
```

US GNP

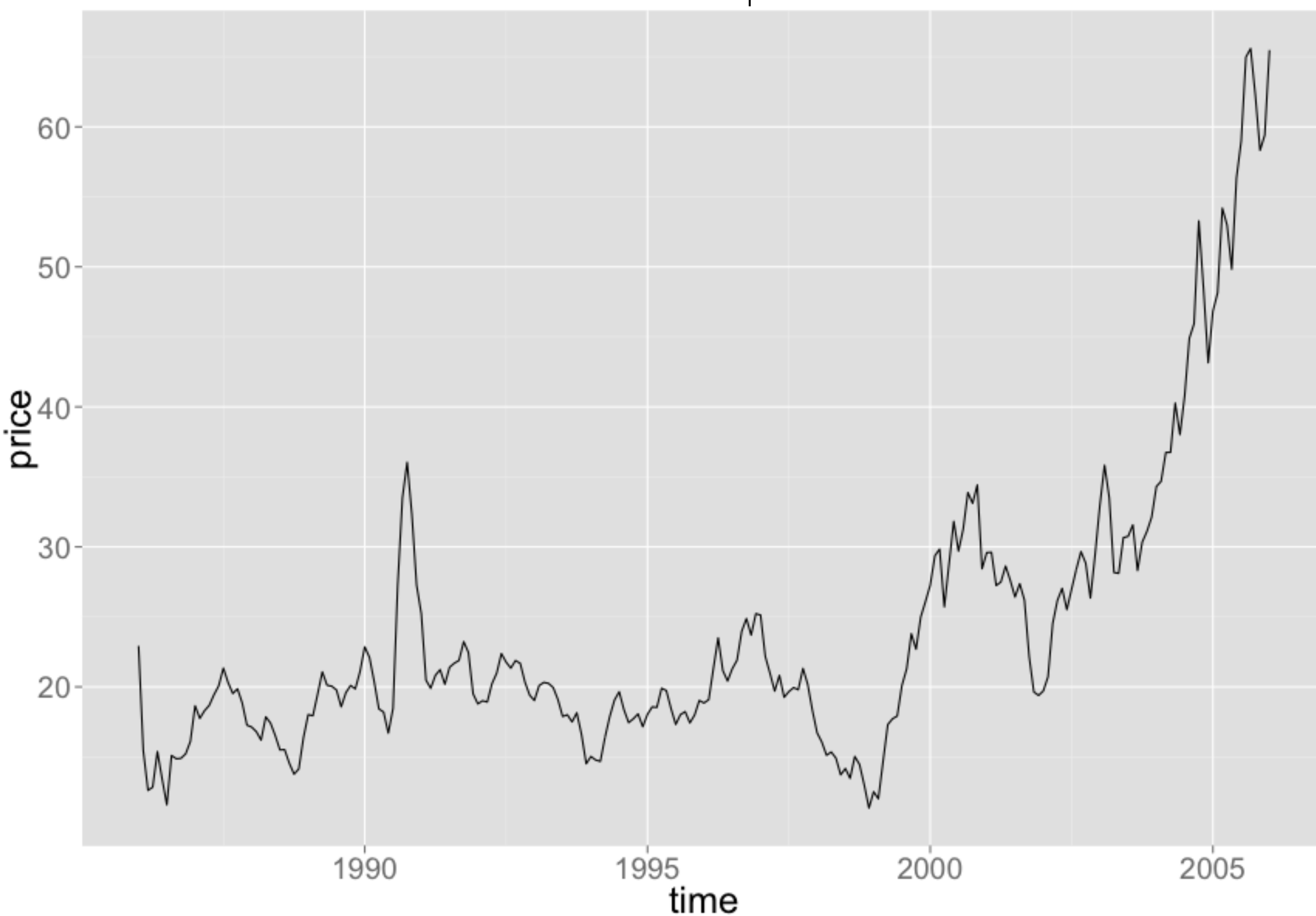
```
load(url("http://www.stat.pitt.edu/stoffer/tsa3/tsa3.rda"))
gnp
```

Sulphur Dioxide (LA county)

```
load(url("http://www.stat.pitt.edu/stoffer/tsa3/tsa3.rda"))
so2
```

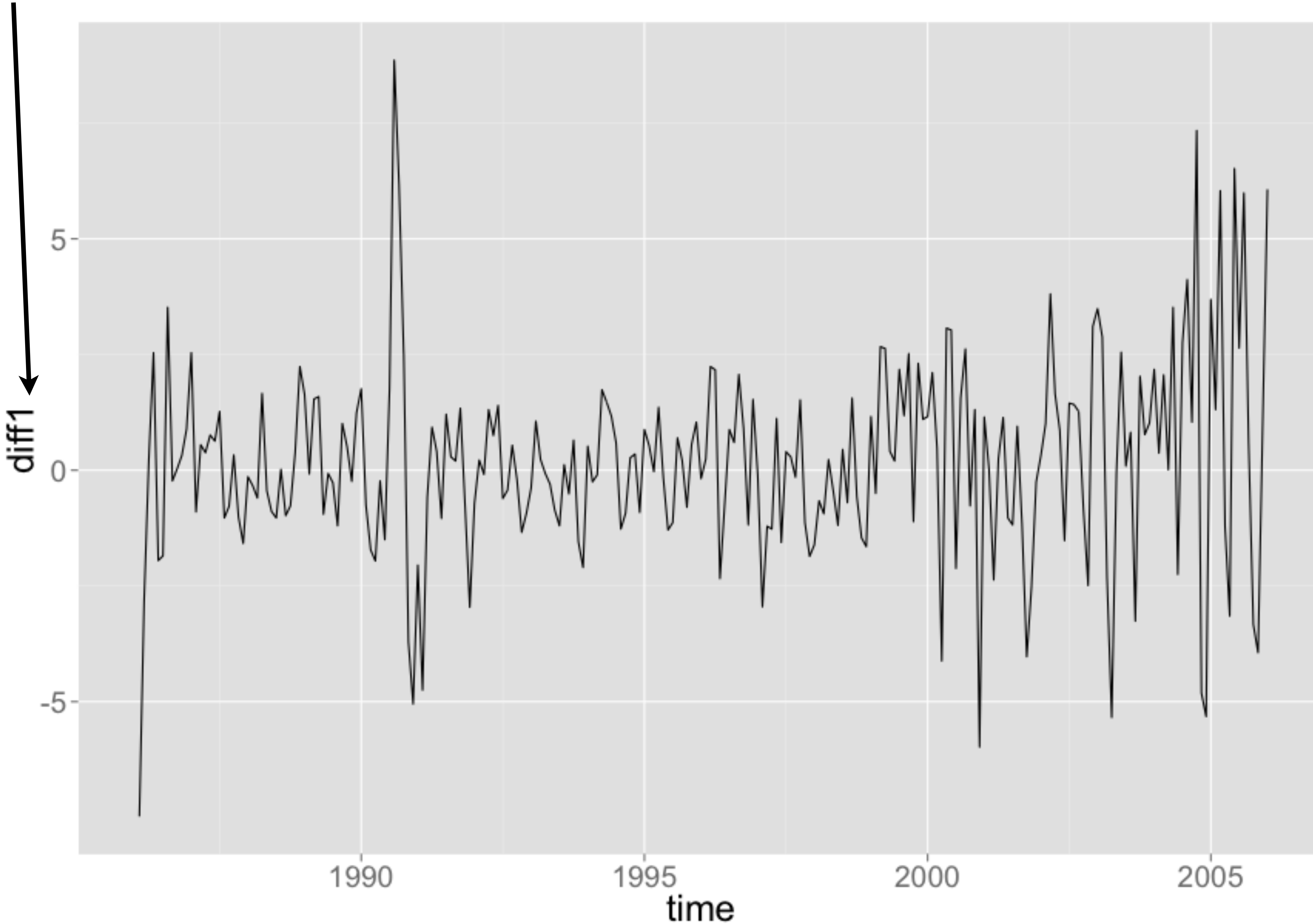
1.

Ex 1 Oil prices

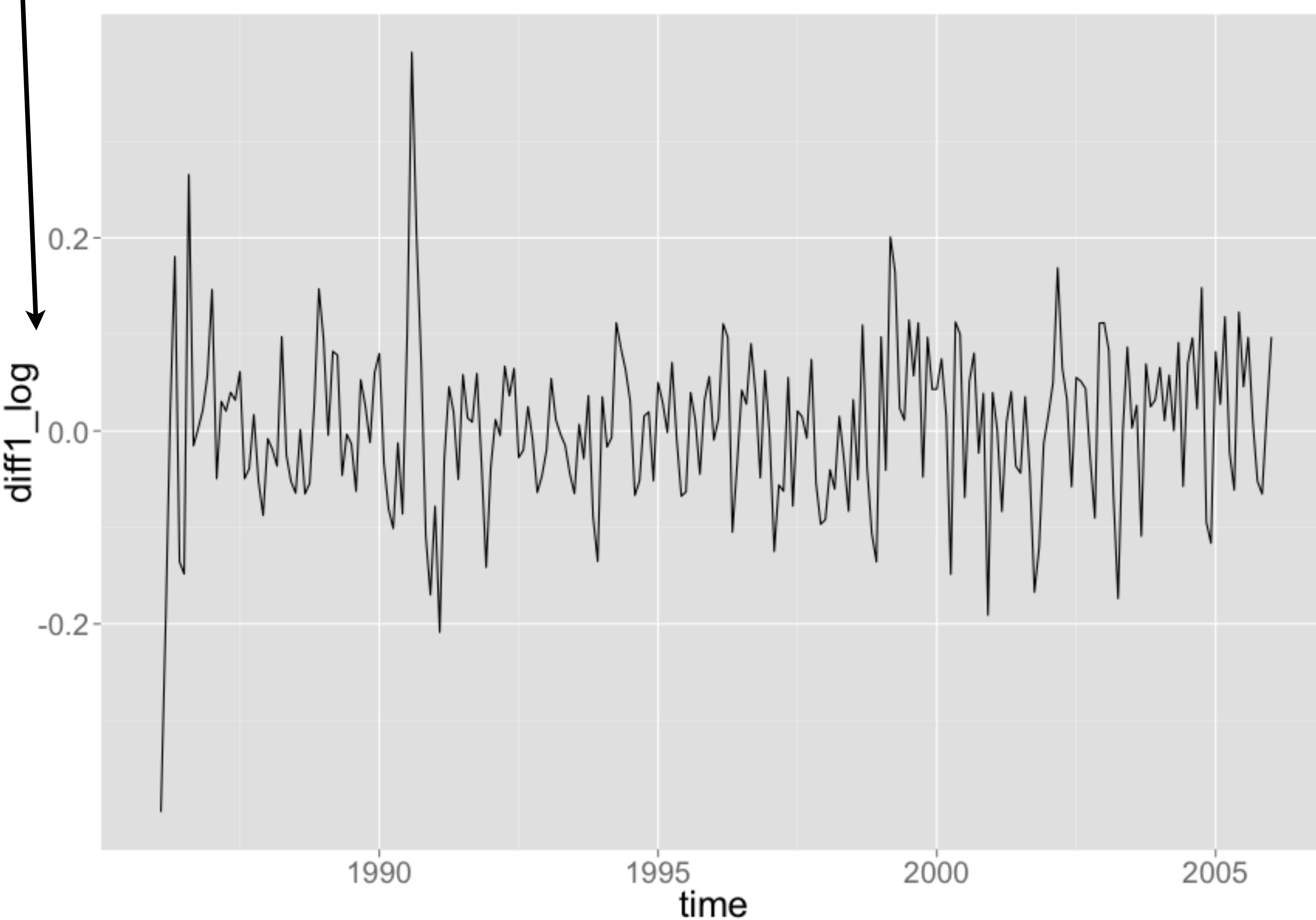


2.

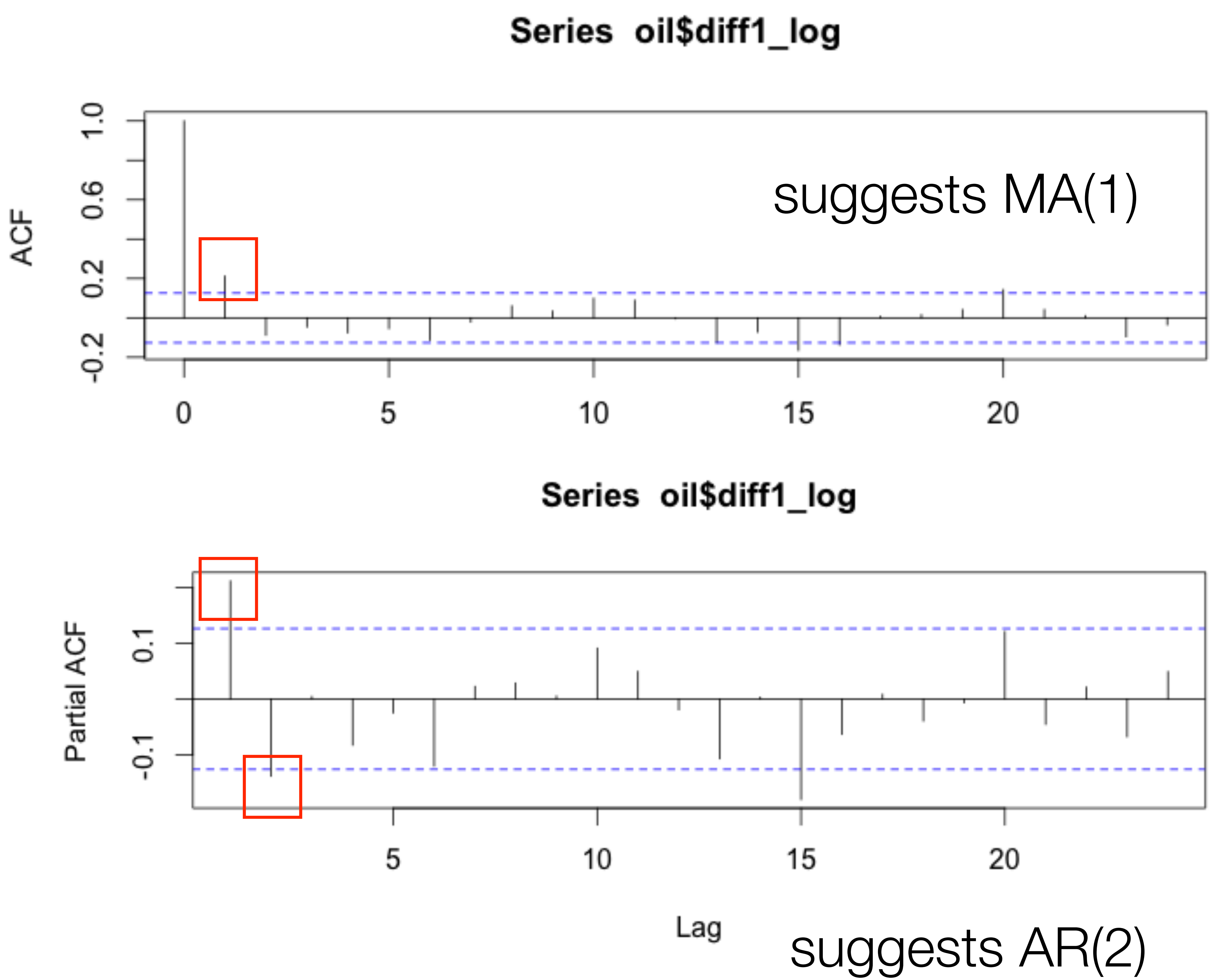
1st difference



2. 1st difference of log(price)



3. ACF and PACF on differenced log price



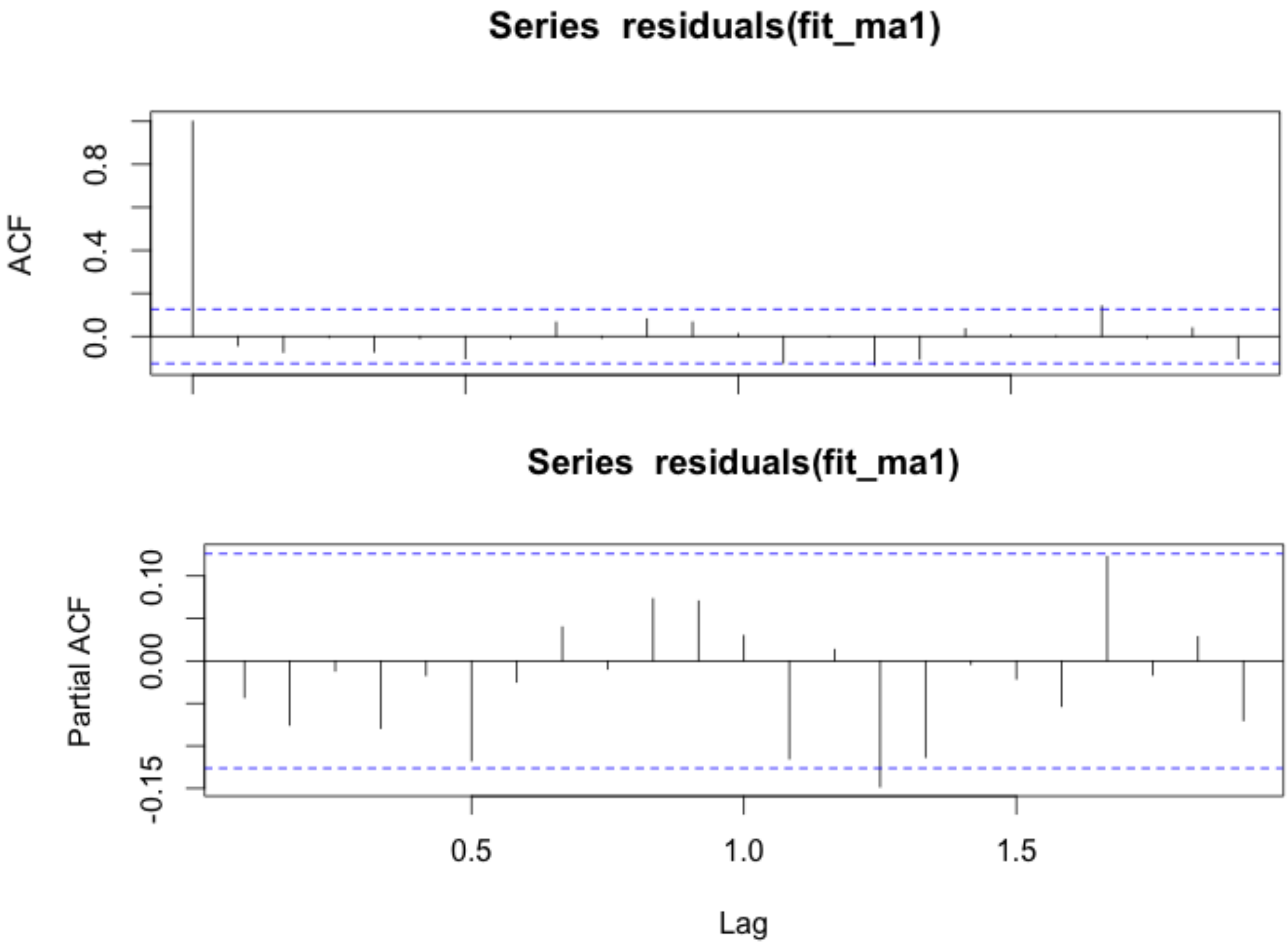
3.

```
n <- length(oil.price)
(fit_ma1 <- arima(log(oil.price), order = c(0, 1, 1), xreg = 1:n))
(fit_ar2 <- arima(log(oil.price), order = c(2, 1, 0), xreg = 1:n))
(fit_arma1 <- arima(log(oil.price), order = c(1, 1, 1), xreg = 1:n))
(fit_ma2 <- arima(log(oil.price), order = c(0, 1, 2), xreg = 1:n))
```

trick ARIMA into estimating a constant in the differenced series

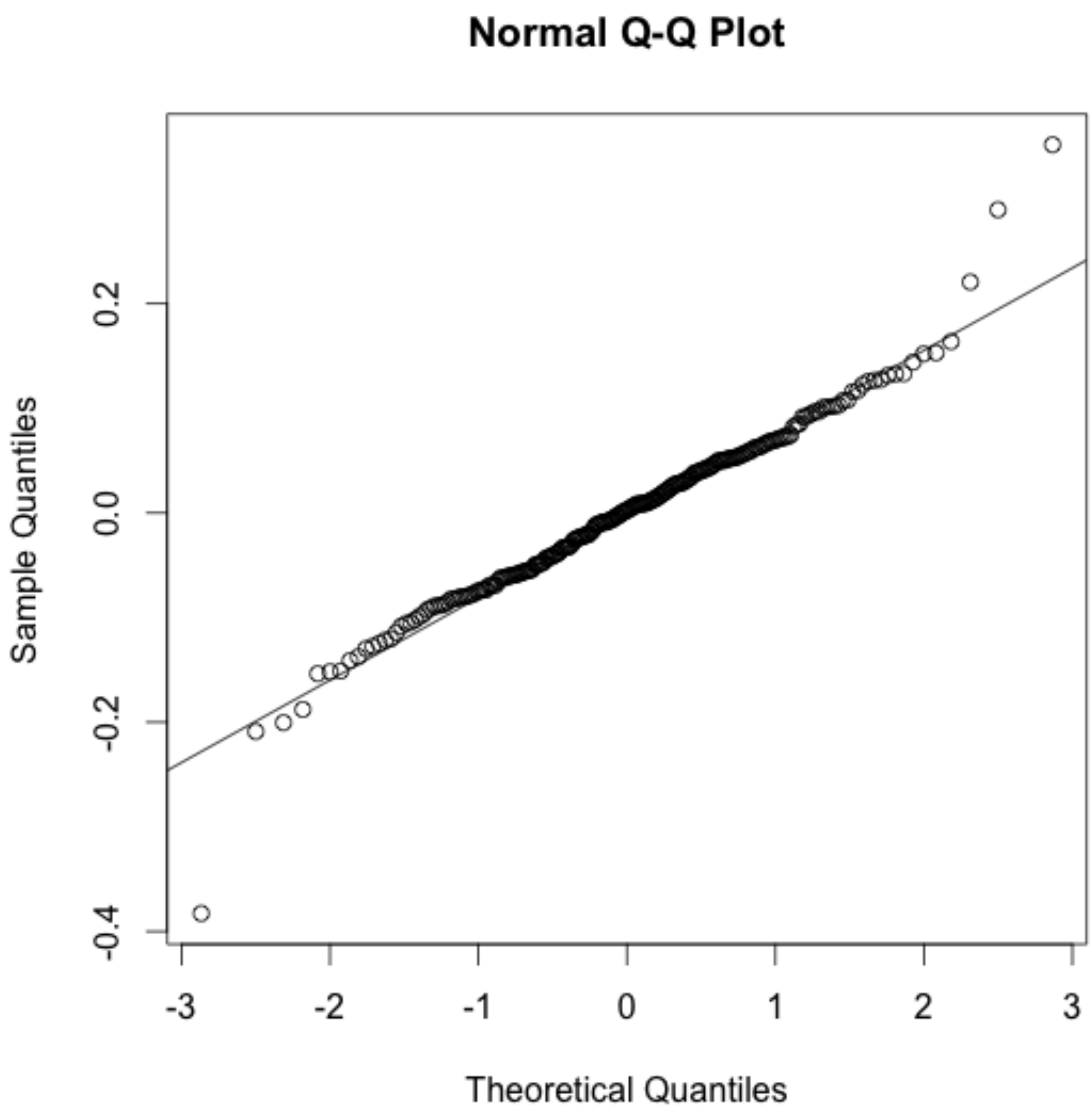
Choose MA(1) based on:
* smallest AIC
* in MA(2) θ_1 is roughly the same and θ_2 isn't significant.

4. ACF and PACF on residuals from MA(1) model



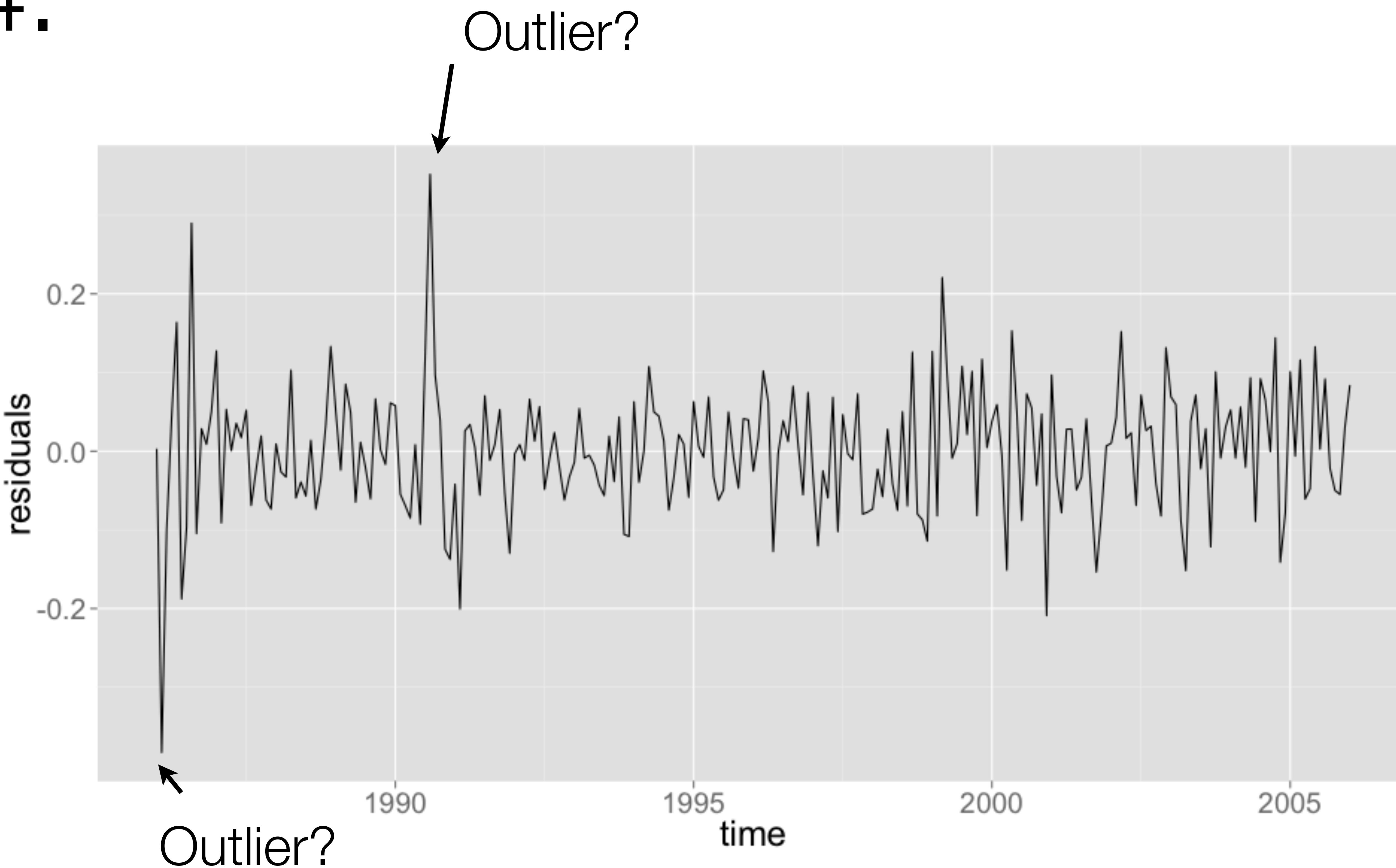
Look good!

4.



Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
-0.4
-0.2
0.0
0.2
-3
-2
-1
0
1
2
3

4.



SARIMA models

I haven't shown you any data with seasonality.

The idea is very similar, if one seasonal cycle lasts for s measurements, then if we difference at lag s,

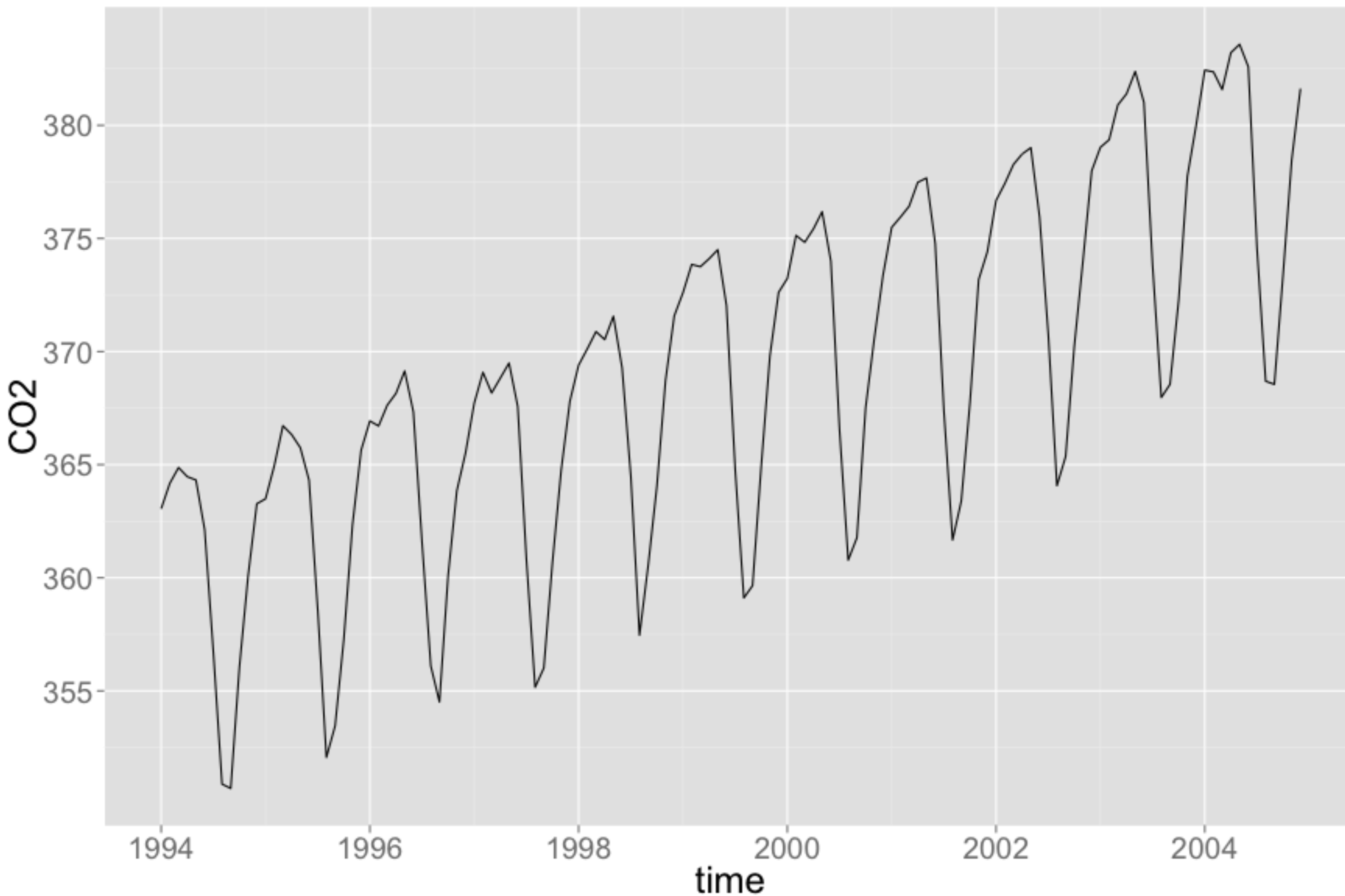
$y_t = \nabla_s x_t = x_t - x_{t-s} = (1 - B^s)x_t$,

we will remove the seasonality.

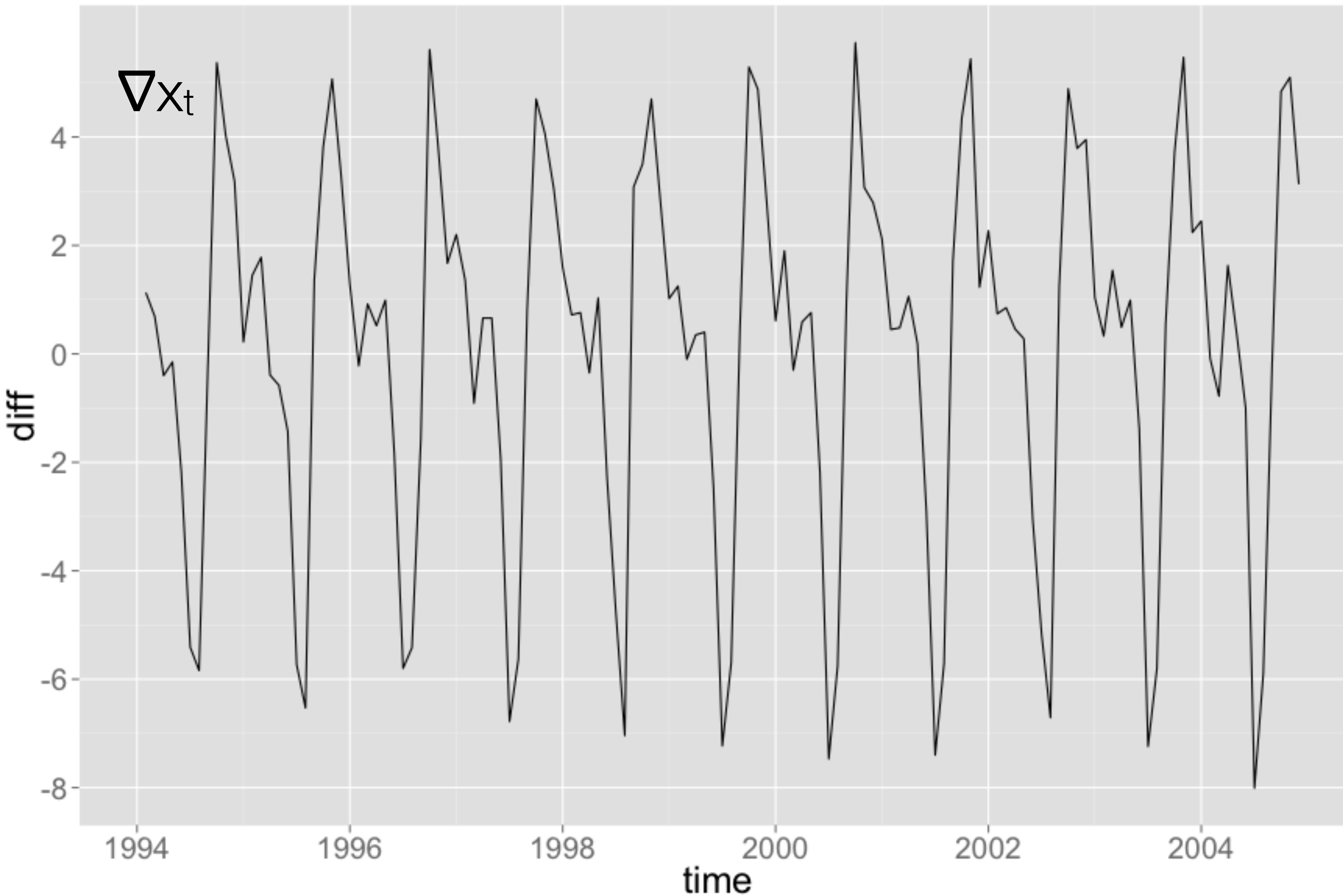
Differencing seasonally D times is denoted,

$\nabla^D{}_s x_t = (1 - B^s)^D x_t$,

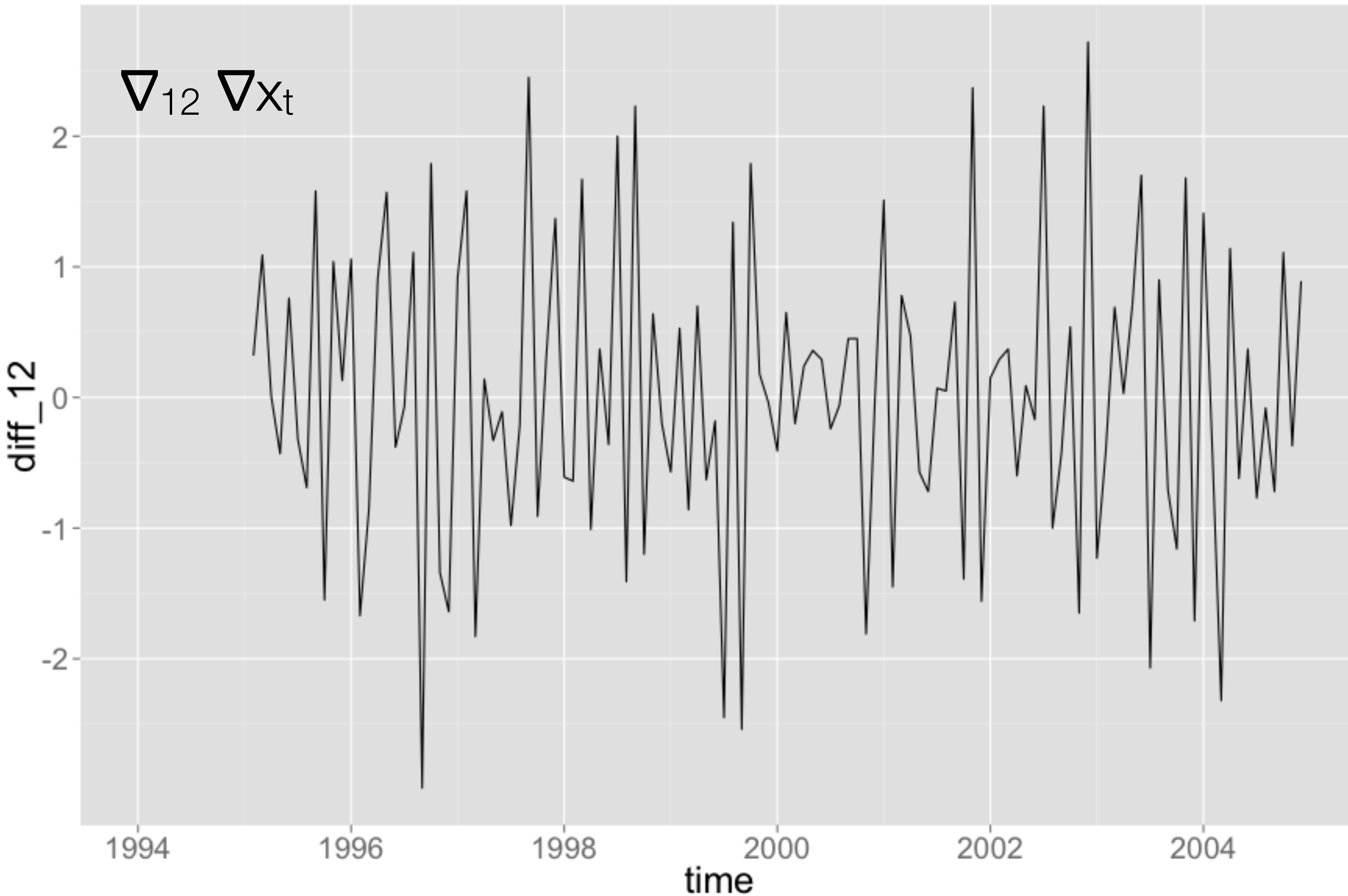
Monthly CO2 level at Alert, Northwest Territories, Canada



First difference



+ first seasonal difference, lag 12



SARIMA

A multiplicative seasonal autoregressive integrated moving average model,

SARIMA(p, d, q) x (P, D, Q)$_s$

is given by

$$\Phi(B^s)\phi(B)\,\nabla^D_s\nabla^d x_t = \Theta(B^s)\theta(B)w_t$$

$\nabla^D_s\nabla^d x_t$ is just an ARMA model with lots of coefficients set to zero.

Have to specify s, then choose p, d, q, P, D and Q

Find model for SARIMA$(1,0,0)\times(0,1,1)_{12}$

Your turn

Find model for SARIMA$(0,1,1) \times (0,1,1)_{12}$

Procedure for **S**ARIMA modeling

We'll assume the primary goal is getting a forecast.

1. Plot the data. Transform? Outliers? Differencing?

2. Difference to remove trend, find d. Then difference to remove seasonality, find D.

3. Examine acf and pacf of differenced series. Find P and Q first, by examining just at lags s, 2s, 3s, etc. Find p and q by examining between seasonal lags.

4. Fit SARIMA$(p, d, q)\times(P, D, Q)_s$ model to original data.

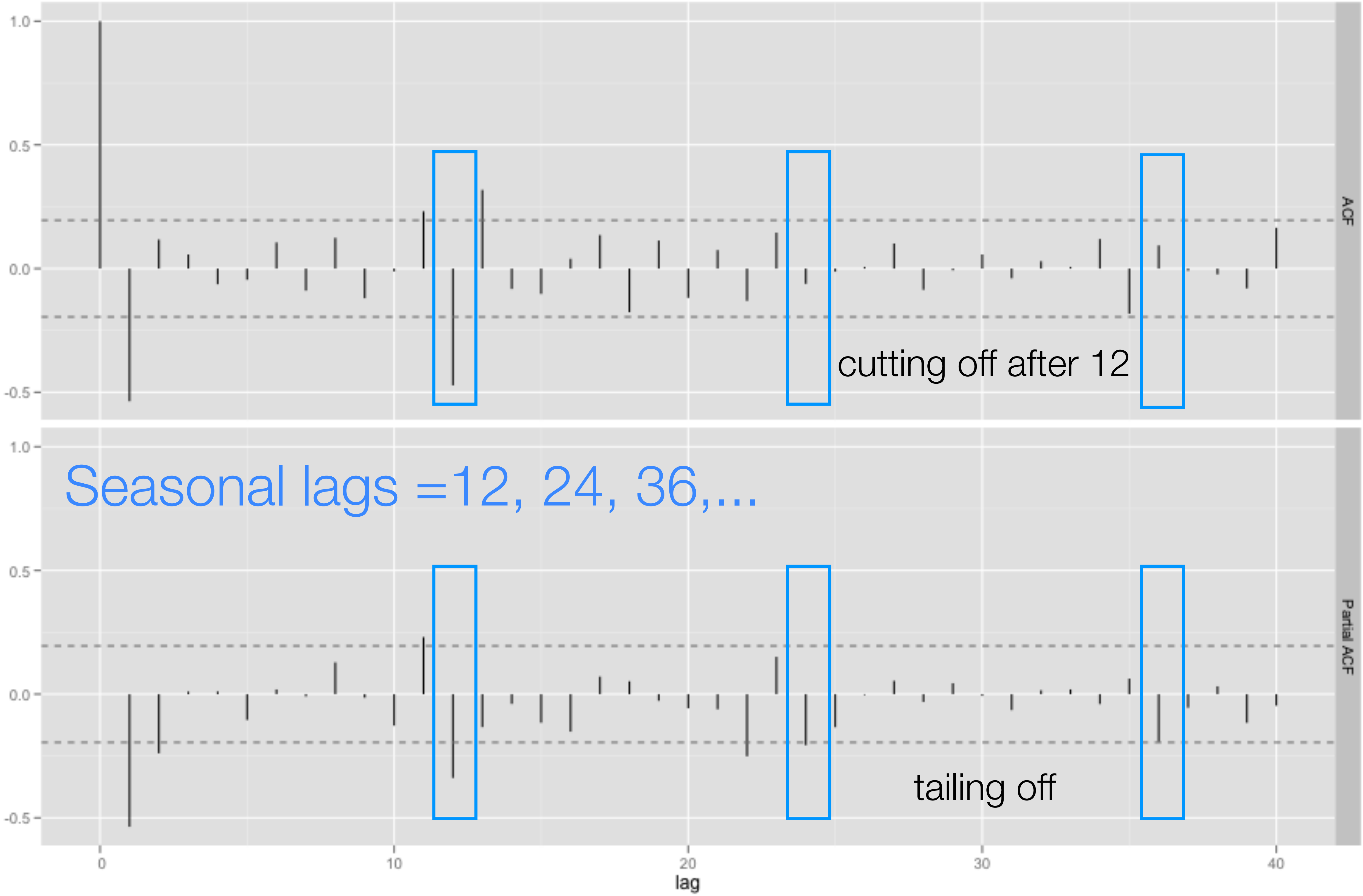
5. Check model diagnostics

6. Forecast (back transform?)

3.

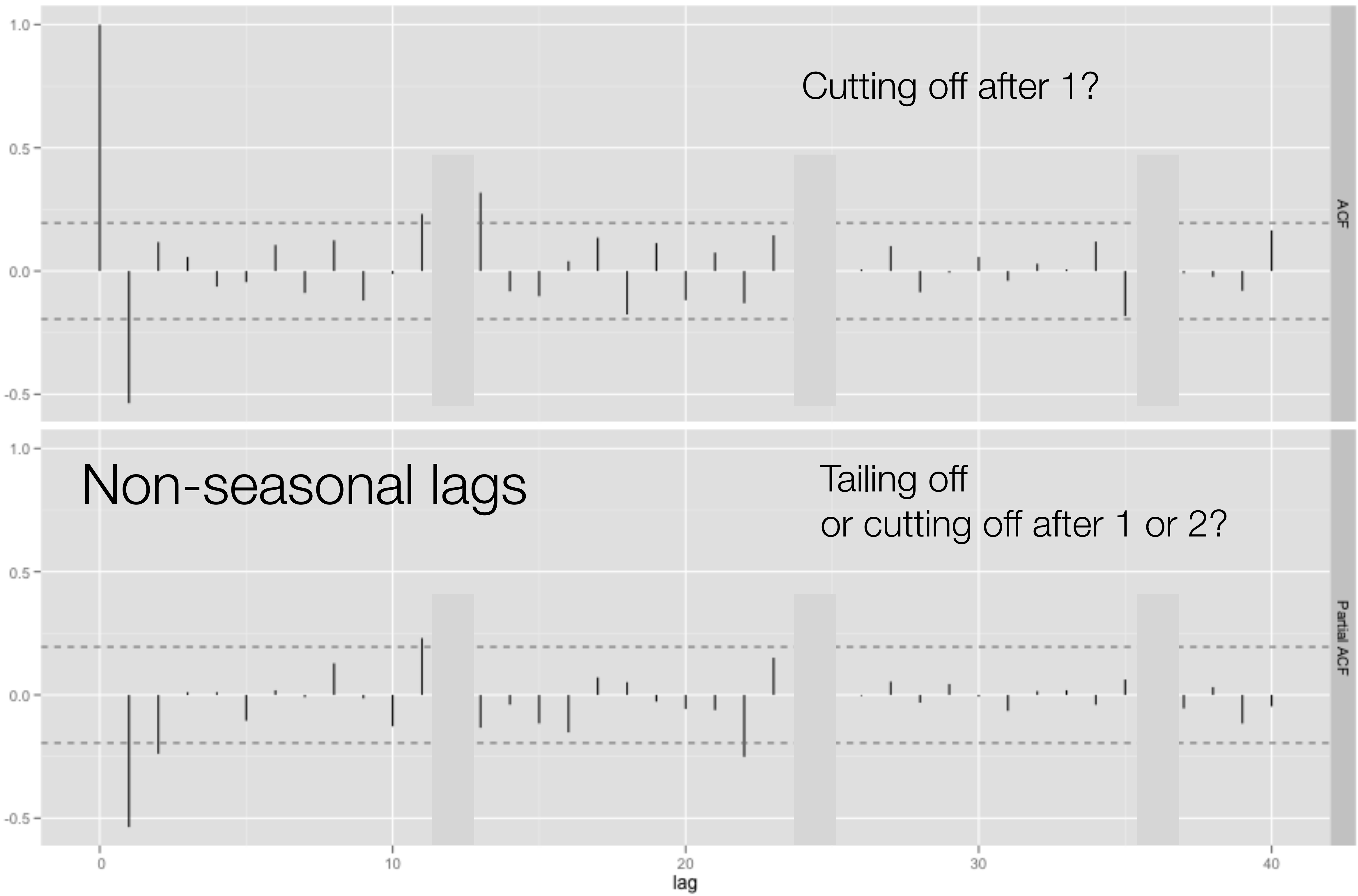
s = 12, D = 1, d = 1

ACF & PACF for $\nabla 12\ \nabla x_t$



3.

s = 12, D = 1, d = 1
ACF & PACF for ∇12 ∇x_t



4.

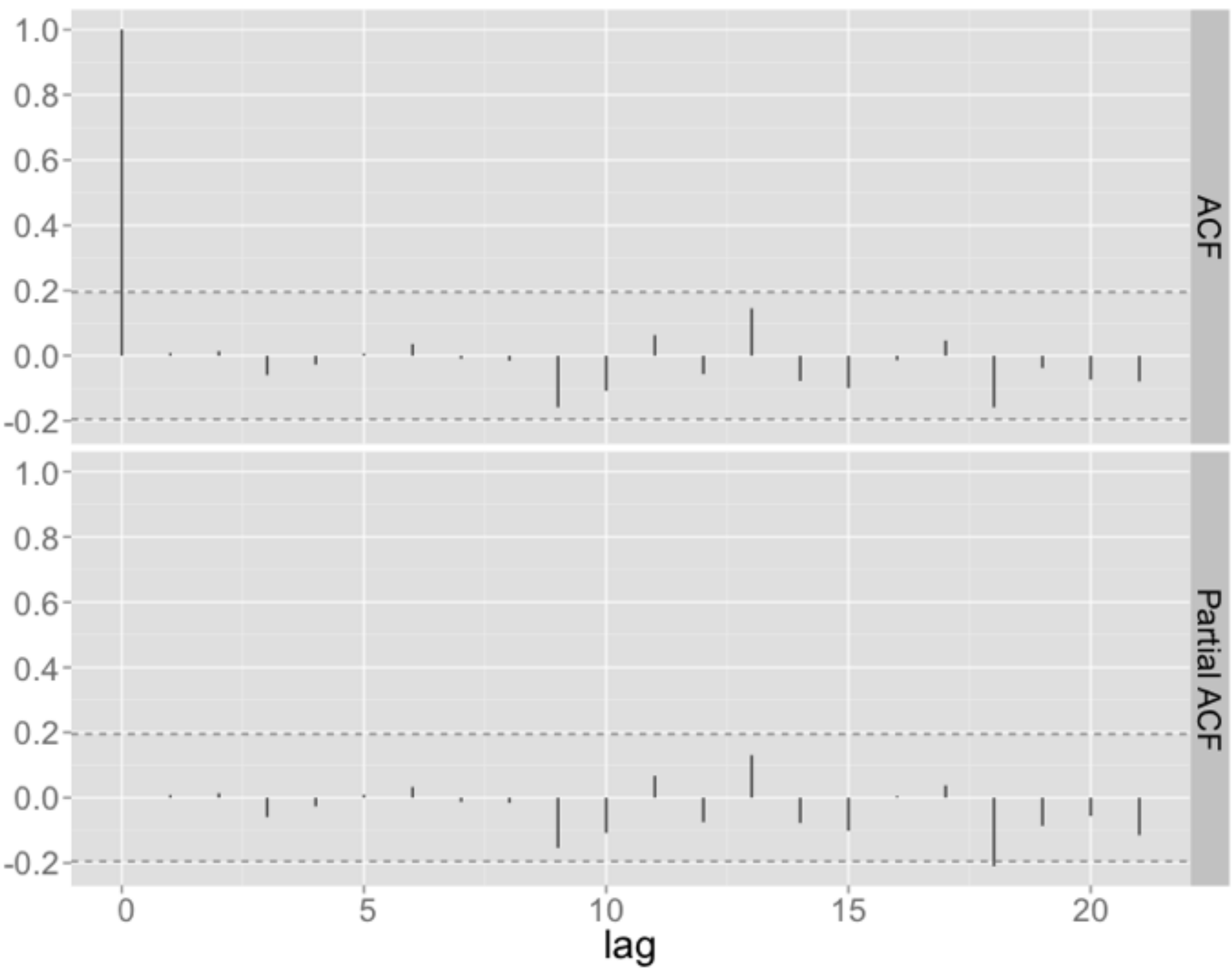
Try

$\boxed{SARIMA\ (0, 1, 1) \times (0, 1, 1)_{12}}$

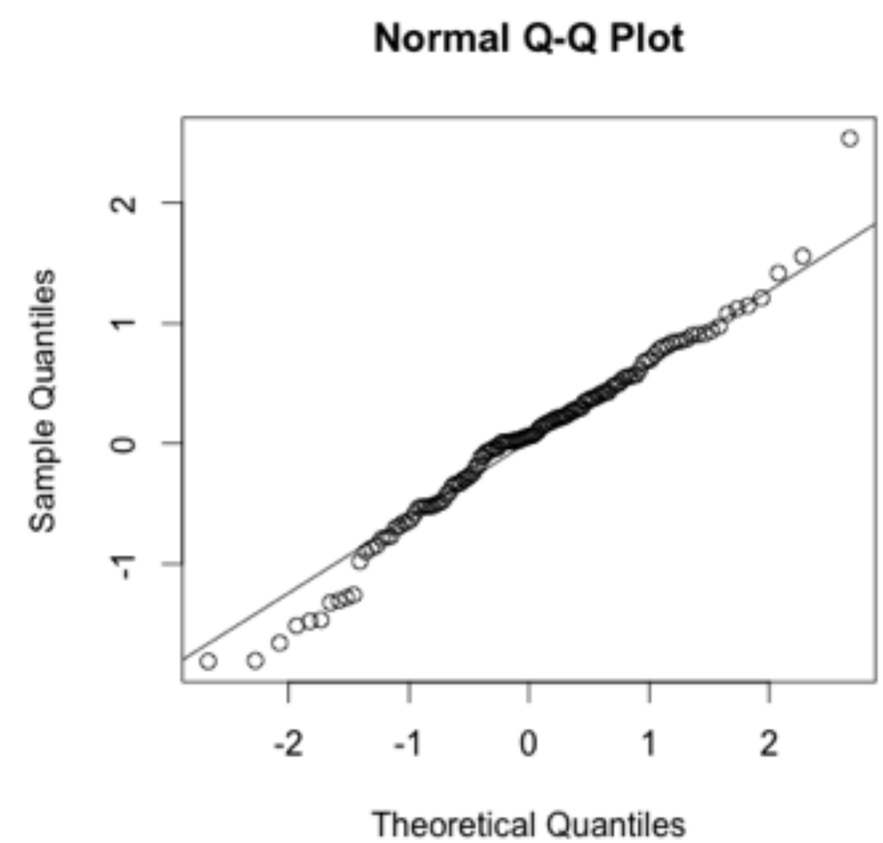
$SARIMA\ (1, 1, 0) \times (0, 1, 1)_{12}$

$SARIMA\ (1, 1, 1) \times (0, 1, 1)_{12}$

5.

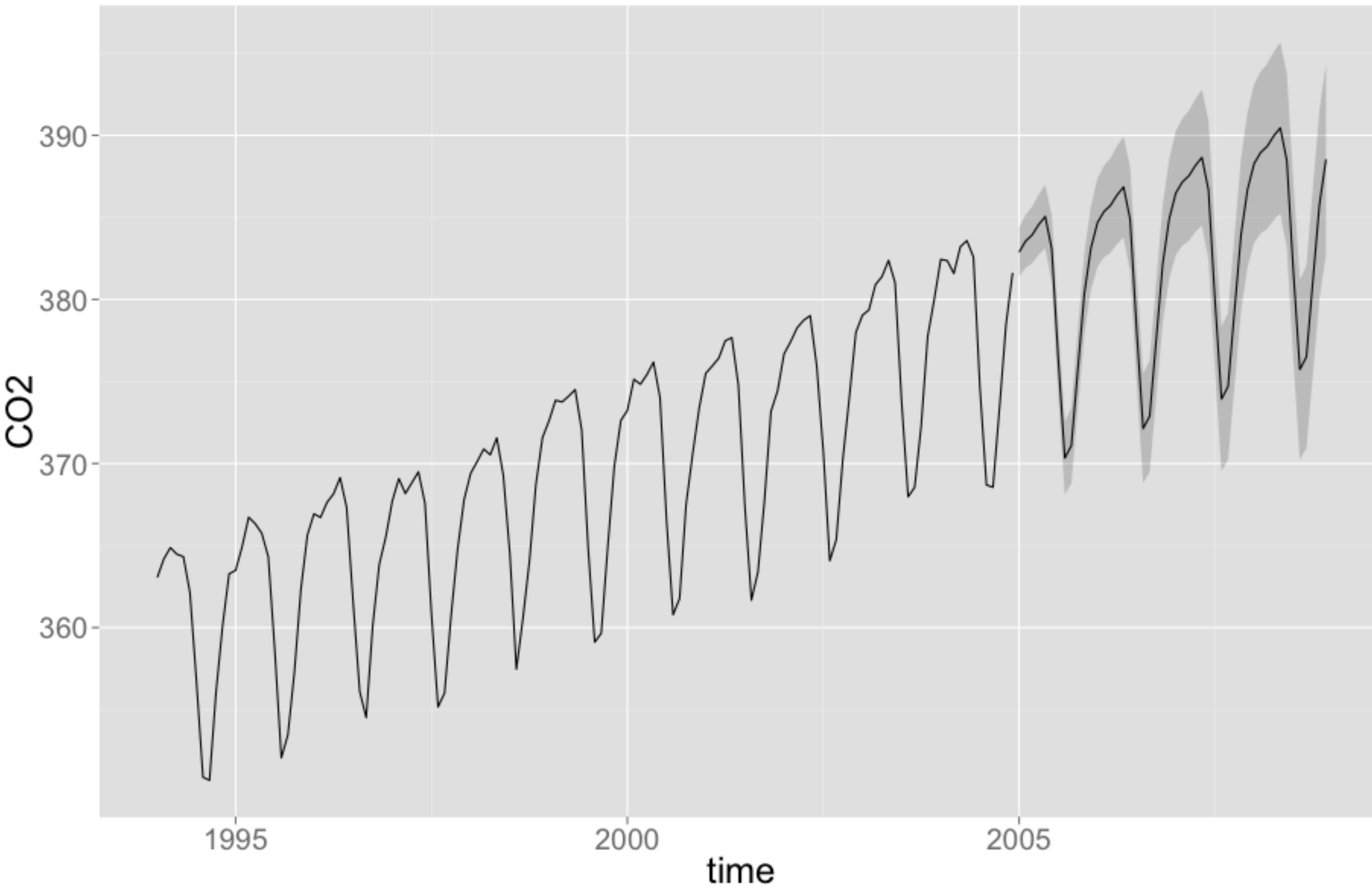


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
ACF
Partial ACF
0
5
10
15
20
lag



Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
-2
-1
0
1
2

6.



Forecasting

Basic Idea: Given an ARMA model, and some past data, we want to predict the future.

Let x_n^{n+k} denote the predictor of x_{n+k} from the values up to x_n.

Technically: We will find a linear function of past values to predict future values that minimizes the prediction mean squared error

(one definition of a good predictor).

Caveats

A forecast relies on the model used for the past also applies in the future.

Today we'll just talk about forecasts for stationary ARMA processes, generally you also want to incorporate trend and seasonality into your forecasts.

Use ARIMA model directly

Plug in zero for future Z_t

Plug in conditional expectation for future X_t.

Plug in observed values for past X_t and Z_t .

Example: predict a SARIMA$(1,0,0) \times (0,1,1)_{12}$ one step ahead

Your turn

What is the one step ahead prediction for an AR(1) process?

Derive predictor

$$x_n^{n+k} = \sigma^2 \sum_{j=k}^{\infty} \psi_j w_{n+k-j}$$

Skip, see Shumway & Stoffer
section 3.5 if interested

BASIC IDEA: use phi form, best
guess for future white noise is zero.

Show error in prediction is

$$\mathrm{Var}(x_n^{n+k}) = \sigma^2 \sum_{j=0}^{k-1} \psi_j^2$$

Skip, see Shumway & Stoffer
section 3.5 if interested

But we don't know $\boldsymbol{\phi}$ and θ?

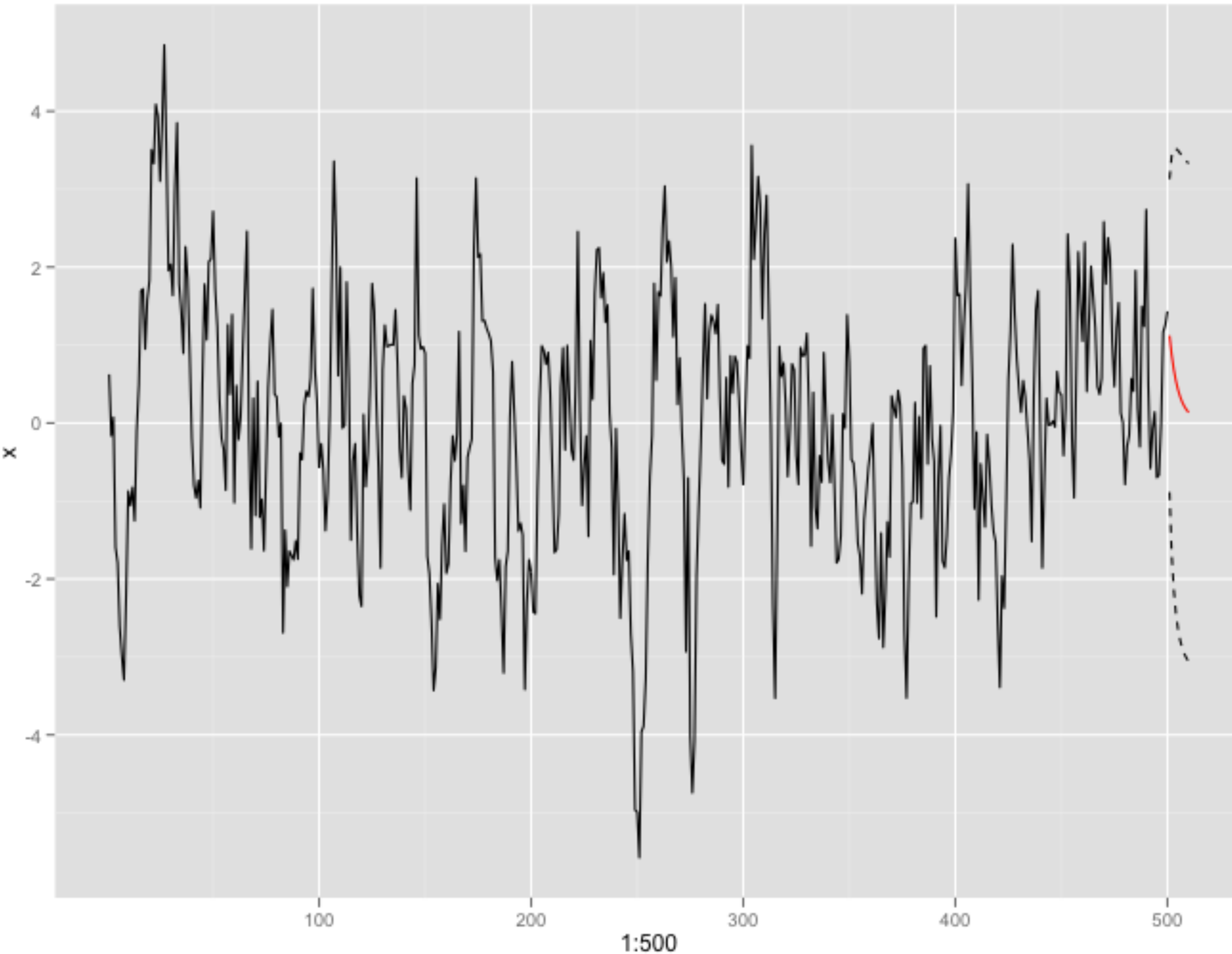
Plug in our estimates and get **approximate** predictions.

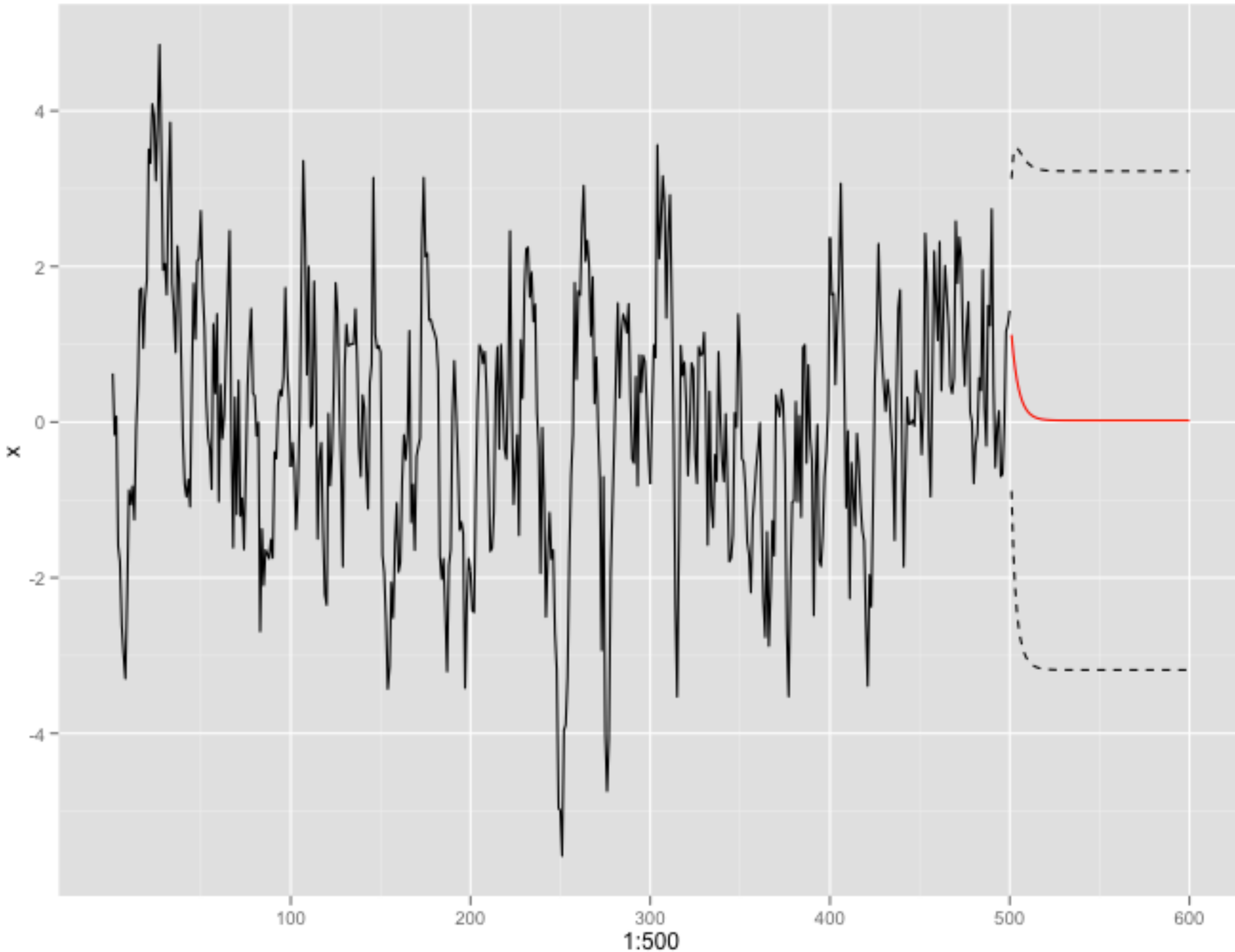
These do not take into account the uncertainty in our estimates.

`predict` in R on an `arima` fit.

```r
x <- arima.sim(model = list(ar = 0.8), 500)
fit_ar1 <- arima(x, order = c(1, 0, 0 ))
predict(fit_ar1, n.ahead = 10)

predict(fit_ar1, n.ahead = 10)
pred.df <- as.data.frame(predict(fit_ar1,
              n.ahead = 10))
```

```r
qplot(1:500, x, geom = "line") +
     geom_line(aes(x = 501:510, pred - 2*se), data = pred.df,
         linetype = "dashed") +
    geom_line(aes(x = 501:510, pred + 2*se), data = pred.df,
         linetype = "dashed") +
    geom_line(aes(x = 501:510, pred), data = pred.df, colour = "red")

pred.df.100 <- as.data.frame(predict(fit_ar1, n.ahead = 100))

qplot(1:500, x, geom = "line") +
     geom_line(aes(x = 501:600, pred - 2*se), data = pred.df.100,
         linetype = "dashed") +
    geom_line(aes(x = 501:600, pred + 2*se), data = pred.df.100,
         linetype = "dashed") +
    geom_line(aes(x = 501:600, pred), data = pred.df.100, colour = "red")
```


4
2
0
-2
-4
x
100
200
300
400
500
1:500



4
2
0
-2
-4
x
100
200
300
400
500
600
1:500