Day 10 - Autoregressive models

Introduction

We continue our discussion of basic stochastic models, introducing the generating autoregressive model of order p.

```
# packages
library(tidyverse)
library(lubridate)
```

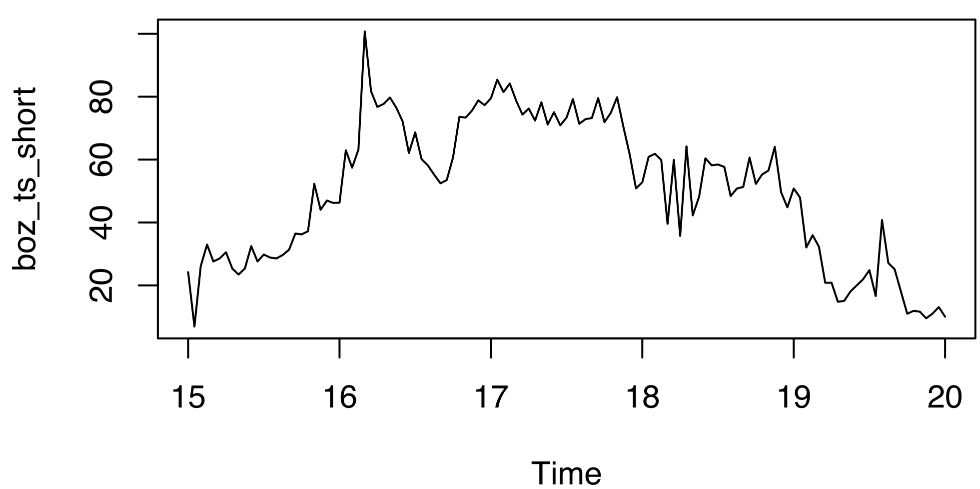
Notes for the exam:

- When I ask to describe the seasonal cycle, talk about when highs & lows occur

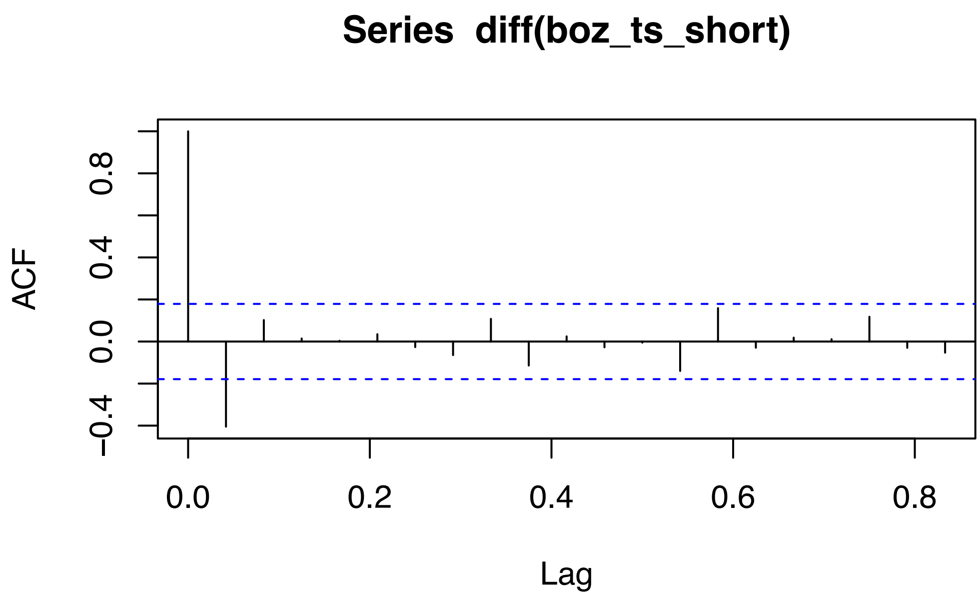
Review

Read in the Bozeman air quality data set, create a time series object, and filter the time series to only observations between September 15^{th} and September 20^{th}. Plot the resulting time series.

```
# load air quality data
bz_air <- readRDS("mt_pm25_sept2020.rds")

# create a ts
mt_pm_clean <- bz_air %>%
  mutate(dt = lubridate::ymd_hms(datetime)) %>%
  dplyr::select(dt, rawvalue, everything()) %>%
  arrange(dt)

# create ts
boz_pm_ts <- ts(
  mt_pm_clean$rawvalue,
  start = c(1, 5),
  end = c(30, 5),
  freq = 24
)

boz_ts_short <-  window(boz_pm_ts, start = c(15, 1), end = c(20, 1))

# plot 09/15 - 09/20
plot(boz_ts_short)
```



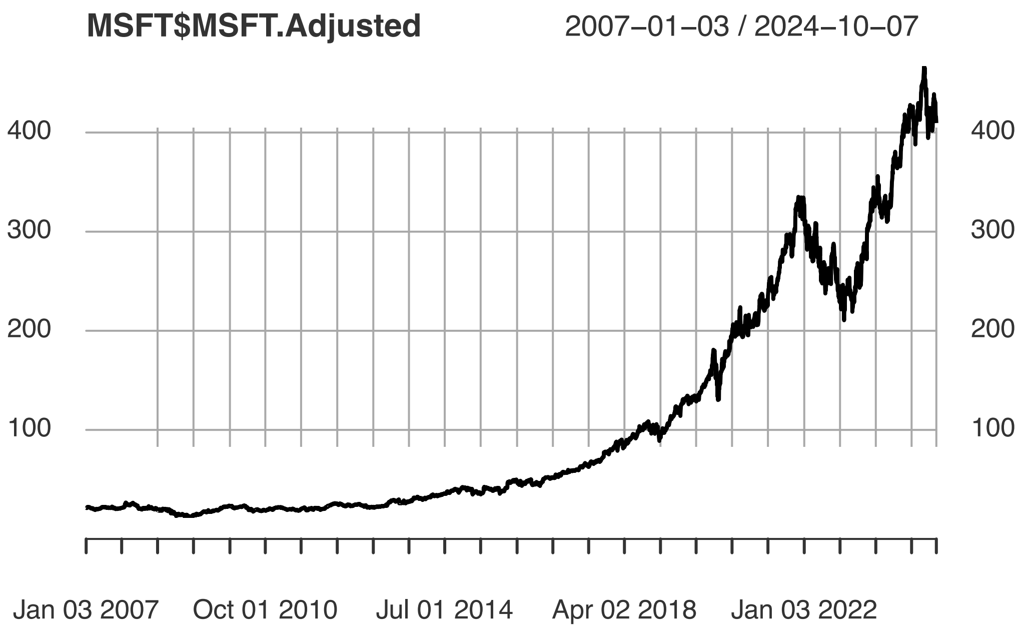
```
# random walk?
acf(diff(boz_ts_short))
```



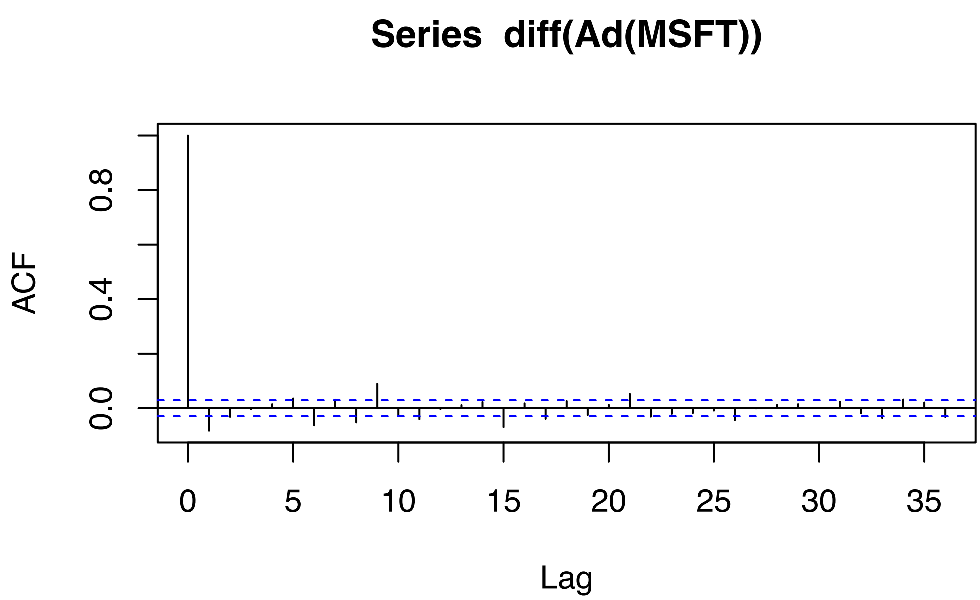

```
# another example
library(quantmod)
getSymbols('MSFT', src = 'yahoo')
```

```
[1] "MSFT"
```

```
plot(MSFT$MSFT.Adjusted)
```




```
acf(diff(Ad(MSFT)), na.action = na.omit)
```



Autoregressive models AR(p)

Note

A time series $\{x_t\}$ is an autoregressive model of order p or AR(p) if

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + w_t$$

where $\{w_t\}$ is white noise and the α_i are the model parameters. It can be shown that the AR(p) model can be expressed as a polynomial of order p in the backshift operator:

$$\theta_p(B)x_t = \underbrace{(1 - \alpha_1 B - \alpha_2 B^2 - \cdots - \alpha_p B^p)}_{\text{characteristic polynomial}}x_t = w_t$$

A few notes on AR(p) models:

- The random walk is a special case of AR(1) with $\alpha_1 = 1$.

  $x_t = x_{t-1} + w_t$ ↳ $x_t = \alpha x_{t-1} + w_t$

- The EWMA (exponentially weighted moving avg) model is a special case of an AR process with $\alpha_i = \alpha(1-\alpha)^i$ as p approaches infinity.

  $x_t = \alpha x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + \alpha(1-\alpha)^3 x_{t-3} + \ldots$

- A prediction at time t is given by

$$\hat{x}_t = \hat{\alpha}_1 x_{t-1} + \hat{\alpha}_2 x_{t-2} + \cdots + \hat{\alpha}_p x_{t-p}$$

- Model parameters are estimated by minimizing the sum of squared error.

Is an AR(p) process stationary?

It depends :)

Note

The equation $\theta_p(B) = 0$ is called the characteristic equation. The roots of the characteristic polynomial may be used to determine whether an AR(p) process is stationary.

If all roots of the characteristic polynomial exceed 1 in magnitude, the model is stationary. You may use the `polyroot` function in R to find the roots of polynomials.

Example: Determine whether the AR(1) model $x_t = \frac{1}{2}x_{t-1} + w_t$ is stationary.

obtain C.P.

$x_t = \frac{1}{2} B x_t + w_t$

$x_t - \frac{1}{2} B x_t = w_t$

$\left(1 - \frac{1}{2} B\right) x_t = w_t$

obtain root:

$1 - \frac{1}{2} B = 0$

$(2)\ 1 = \frac{1}{2} B\ (2)$

$B = 2$

Example: Determine whether the AR(2) model $x_t = x_{t-1} - \frac{1}{4}x_{t-2} + w_t$ is stationary.

Example: Determine whether the AR(2) model $x_t = \frac{1}{2}x_{t-1} - \frac{1}{2}x_{t-2} + w_t$ is stationary.

AR(1) processes

A time series x_t is an AR(1) process if

$$x_t = \alpha x_{t-1} + \omega_t$$

The second-order properties of an AR(1) process are:

$$\mu(t) = 0$$

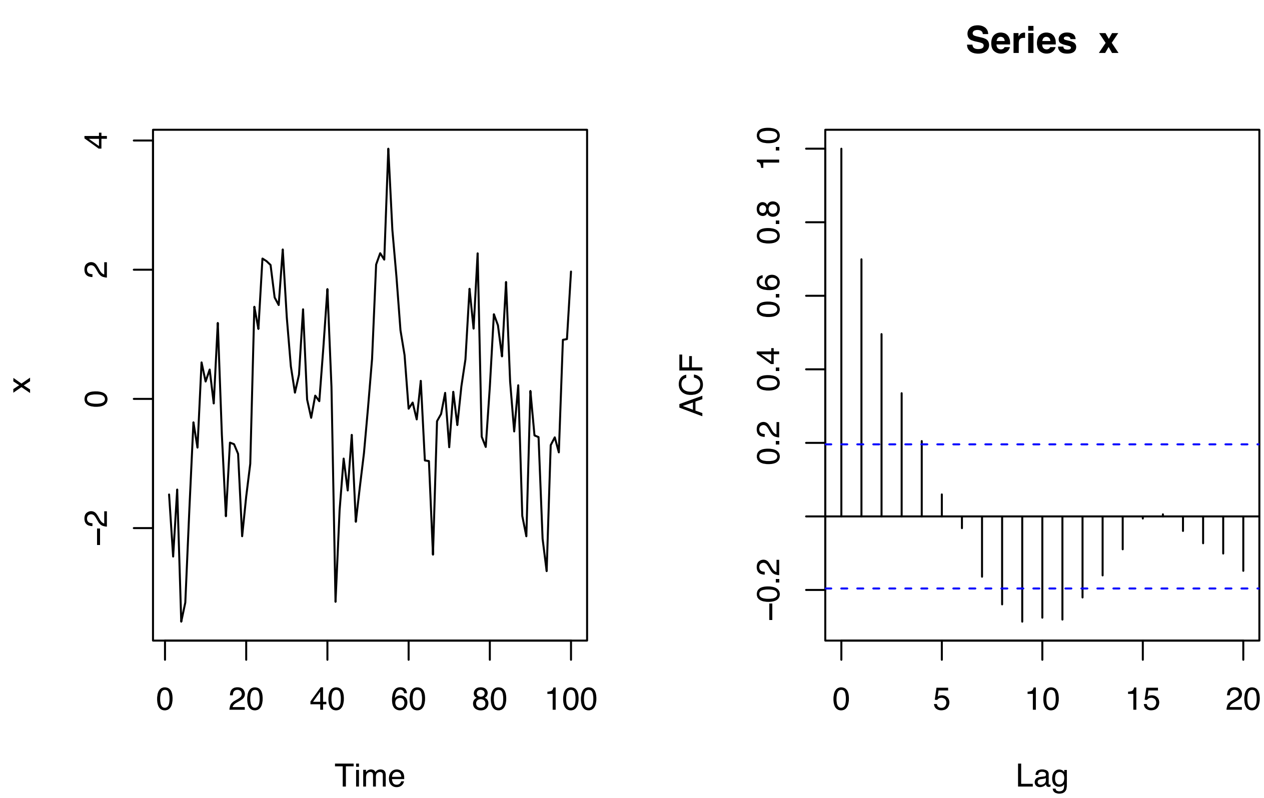
$$\gamma_k = \frac{\alpha^k \sigma^2}{(1-\alpha^2)}$$

$$\rho_k = \alpha^k$$

How can we simulate an AR(1) process?

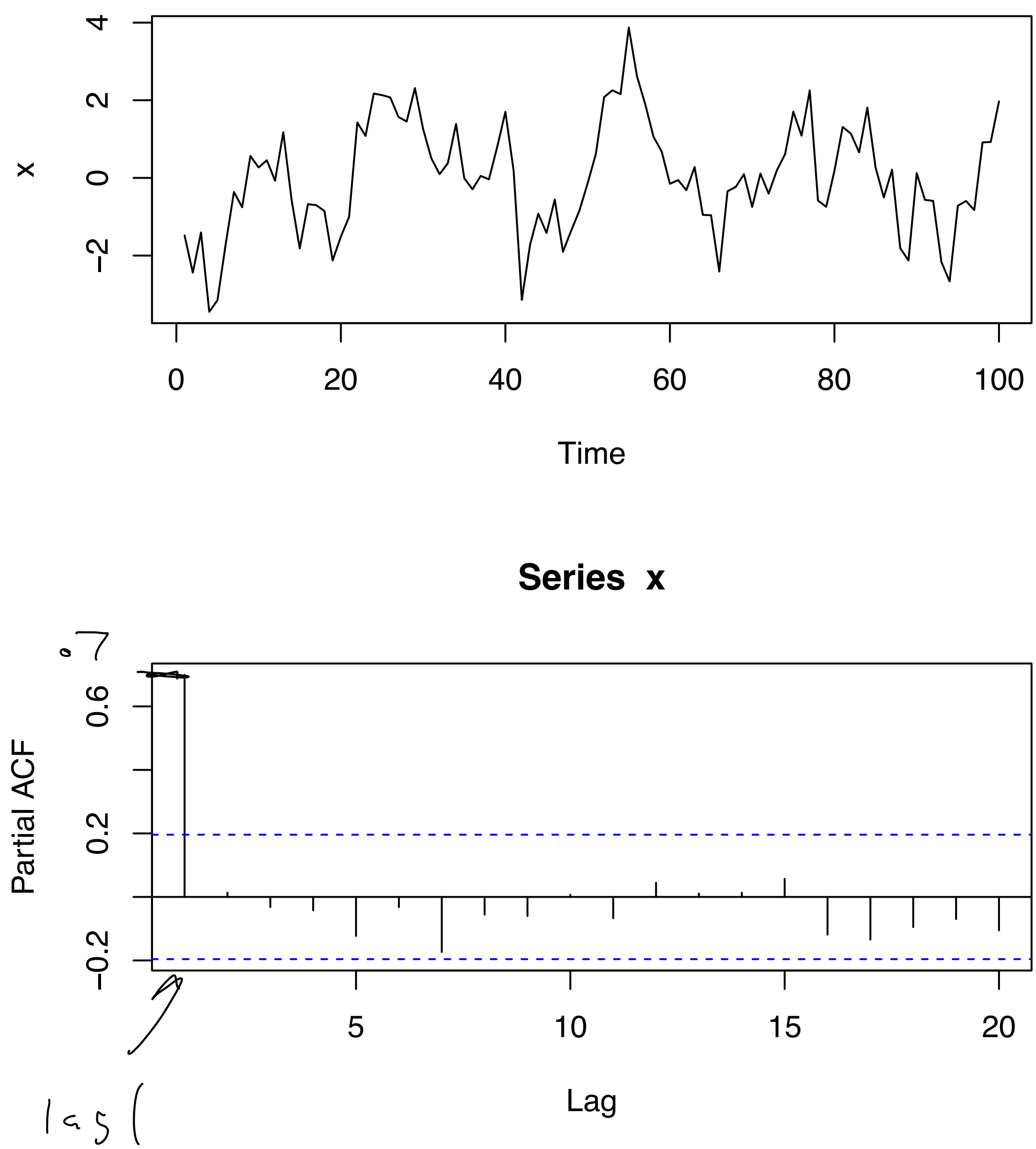
α

```
set.seed(10062024)
x <- w <- rnorm(100)
for(t in 2:100) x[t] <- 0.7 * x[t-1] + w[t]
par(mfrow = c(1, 2))
plot(x, type = "l", xlab = "Time")
acf(x)
```



> **Note**
>
> The partial ACF at lag k is the correlation that results after removing the effect of any correlation due to terms at shorter lags.

```
par(mfrow = c(2, 1))
plot(x, type = "l", xlab = "Time")
pacf(x)
```



Fitting an AR(p) process

To fit an AR(p) process, we use the `ar` function in R. To select the order of the AR process, R minimizes the AIC.

$$AIC = 2 \cdot (-\text{log-likelihood} + \text{number of parameters})$$

```
# a few things
par(mfrow = c(1,1))
ar_fit <- ar(x)
str(ar_fit)
```

$\hat{\alpha}_1$

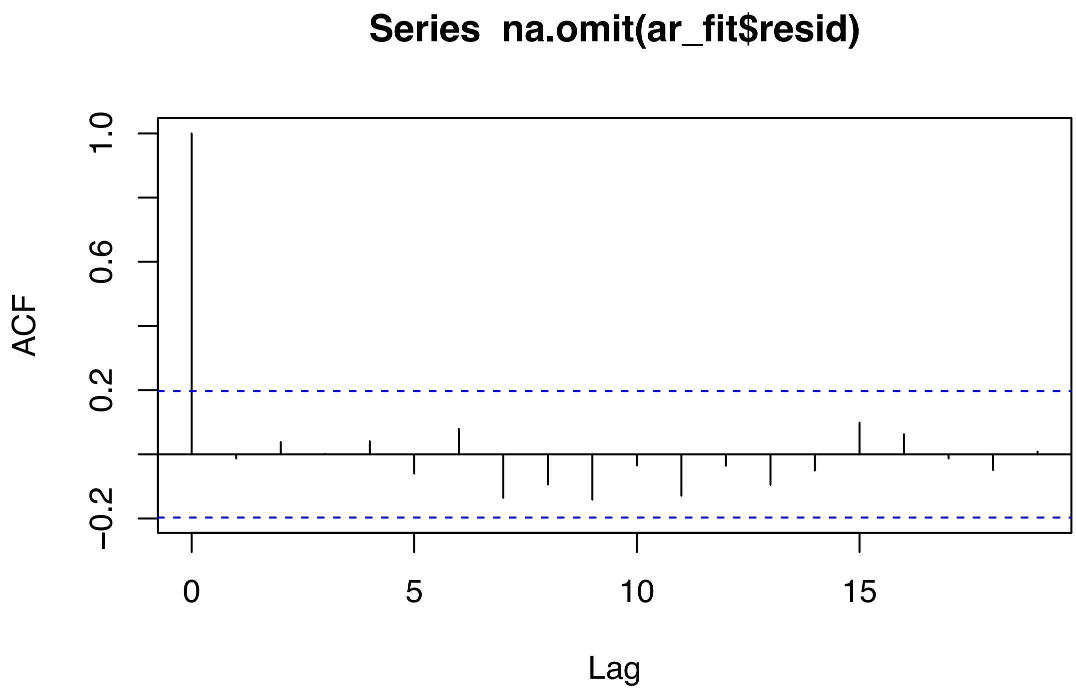
```
List of 15
 $ order       : int 1
 $ ar          : num 0.699
 $ var.pred    : num 1.06
 $ x.mean      : num -0.00725
 $ aic         : Named num [1:21] 65.14 0 1.98 3.88 5.7 ...
  ..- attr(*, "names")= chr [1:21] "0" "1" "2" "3" ...
 $ n.used      : int 100
 $ n.obs       : int 100
 $ order.max   : num 20
 $ partialacf  : num [1:20, 1, 1] 0.6993 0.0133 -0.032 -0.0423 -0.123 ...
 $ resid       : num [1:100] NA -1.404 0.306 -2.466 -0.738 ...
 $ method      : chr "Yule-Walker"
 $ series      : chr "x"
 $ frequency   : num 1
 $ call        : language ar(x = x)
 $ asy.var.coef: num [1, 1] 0.00521
 - attr(*, "class")= chr "ar"
```

smallest Δ AIC

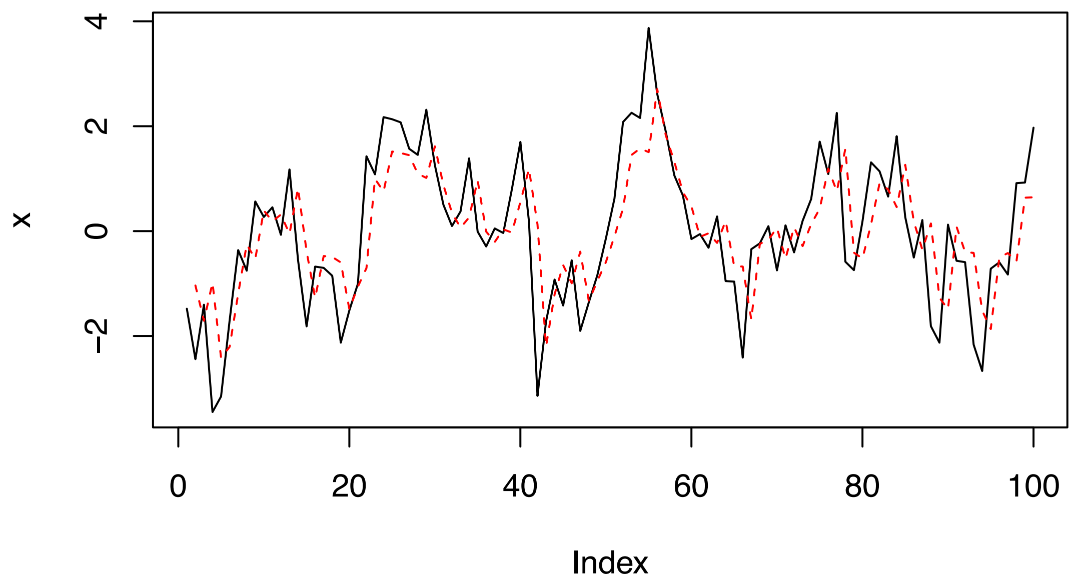
```
ar_fit$aic
```

```
        0         1         2         3         4         5         6         7 
65.141818  0.000000  1.982267  3.879878  5.700571  6.175876  8.072266  7.017330 
        8         9        10        11        12        13        14        15 
 8.705460 10.340195 12.335376 13.885613 15.686569 17.675725 19.657472 21.333858 
       16        17        18        19        20 
21.913735 22.088854 23.184022 24.699330 25.586472 
```

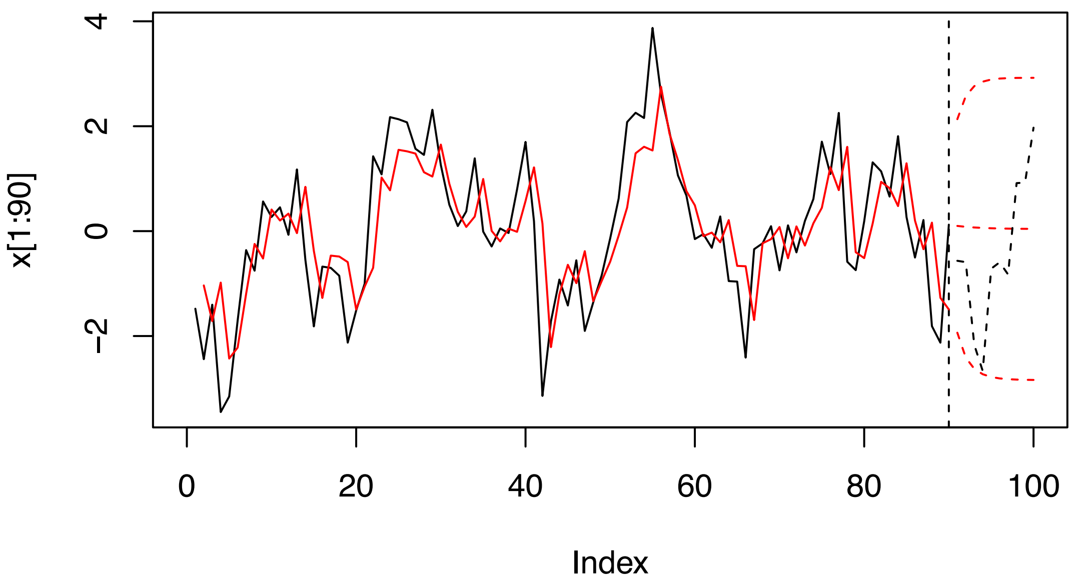
```
acf(na.omit(ar_fit$resid))
```




```
# obtain predictions 2:N for AR(1) model
fitted <- ar_fit$x.mean + ar_fit$ar * (x[1:(length(x)-1)] - ar_fit$x.mean)
plot(x, type = "l")
lines(fitted ~ c(2:100), col = "red", lty = 2)
```




```
# what about forecasts?
## pretend we observe 1:90, prediction 91:100
x_short <- x[1:90]
ar_fit_short <- ar(x_short)
fitted <- ar_fit_short$x.mean +
  ar_fit_short$ar *
  (x_short[1:(length(x_short)-1)] - ar_fit_short$x.mean)

plot(x[1:90], xlim = c(0, 100), type ="l")
abline(v = 90, lty = 2)
lines(fitted ~ c(2:90), col = "red")

x_pred <- predict(ar_fit_short, n.ahead = 10)
lines(x_pred$pred, col = "red", lty = 2)
lines(x_pred$pred - 2*x_pred$se, col = "red", lty = 2)
lines(x_pred$pred + 2*x_pred$se, col = "red", lty = 2)
lines(x[91:100] ~ c(91:100), lty = 2)
```


4
2
0
−2
x[1:90]
0
20
40
60
80
100
Index

Closing remarks

- The stochastic models discussed this week (white noise, random walks, and AR(p) processes) are not very useful for forecasting on their own, and are unlikely to compete with procedures like Holt-Winters.
- In practice, these stochastic models are combined with other techniques (regression, moving average, and integrated moving averages) to construct powerful forcasting techniques.
- The AR(p) process is one component of the ARIMA model, which we will discuss towards the end of the semester.