

Modeling and Analysis of Time Series Data

Chapter 3: Stationarity, white noise, and some basic time series models

Edward L. Ionides

Outline

- 1 Concepts of stationarity
- 2 White noise
- 3 Building models using white noise
- 4 Modeling financial returns using white noise

Definition: Weak and strong stationarity

均值始终相同,协方差只和lag有关(都和时间无关)

- A time series model which is both mean stationary and covariance stationary is **weakly stationary** or **second order stationary**.
- A time series model for which all **joint distributions** are invariant to shifts in time is **strongly stationary** or **strictly stationary**.
- Formally, this means that for any collection of times (t_1, t_2, \dots, t_K) , the joint distribution of observations at these times should be the same as the joint distribution at $(t_1 + \tau, t_2 + \tau, \dots, t_K + \tau)$ **for any τ** .
- For equally spaced observations, this becomes: for any collection of timepoints n_1, \dots, n_K , and for any lag h , the joint density function of $(Y_{n_1}, Y_{n_2}, \dots, Y_{n_K})$ is the same as the joint density function of $(Y_{n_1+h}, Y_{n_2+h}, \dots, Y_{n_K+h})$.
任意连续K个数据点联合分布相同

- In our general notation for densities, this strict stationarity requirement can be written as

$$f_{Y_{n_1}, Y_{n_2}, \dots, Y_{n_K}}(y_1, y_2, \dots, y_K) \quad (1)$$

$$= f_{Y_{n_1+h}, Y_{n_2+h}, \dots, Y_{n_K+h}}(y_1, y_2, \dots, y_K). \quad (2)$$

- Strict stationarity implies weak stationarity (check this).

Question 3.1. How could we assess whether a weak stationary model is appropriate for a time series dataset?

1. Plot the data and ACF over different time intervals
2. 如果是弱平稳,则ACF应该相同

Question 3.2. How could we assess whether a strictly stationary model is appropriate for a time series dataset?

Very hard,几乎不可能通过样本数据检验

Question 3.3. Is it usual for time series to be well modeled as stationary (either weakly or strictly)?

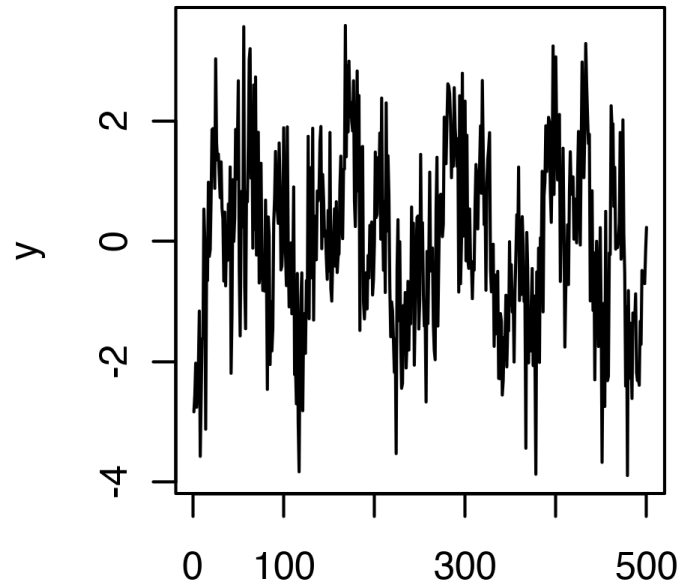
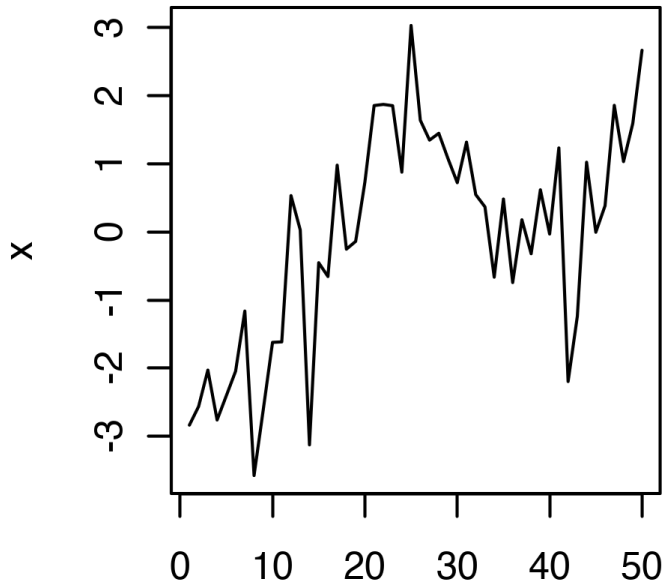
大部分时序数据都有趋势或者不平稳

Question 3.4. If data often do not show stationary behavior, why do many fundamental models have stationarity?

- 稳定模型便于应用和估计
- 可以把不平稳的数据转化为平稳数据

Question 3.5. Is a stationary model appropriate for either (or both) the time series below? Explain.

- X看起来不平稳,Y看起来平稳
- 但Y是X的前50个点
- 平稳性和Time Scale有关



White noise

A time series model $\epsilon_{1:N}$ which is weakly stationary with

$$\begin{aligned}\mathbb{E}[\epsilon_n] &= 0 \\ \text{Cov}(\epsilon_m, \epsilon_n) &= \begin{cases} \sigma^2, & \text{if } m = n \\ 0, & \text{if } m \neq n \end{cases},\end{aligned}$$

is said to be **white noise** with variance σ^2 . 只要是均值为0的i.i.d都可以

- The “noise” is because there’s no pattern, just random variation. If you listened to a realization of white noise as an audio file, you would hear a static sound.
- The “white” is because all frequencies are equally represented. This will become clear when we do frequency domain analysis of time series.
- Signal processing—sending and receiving signals on noisy channels—was a motivation for early time series analysis.

Example: Gaussian white noise

In time series analysis, a sequence of independent **identically** distributed (iid) Normal random variables with mean zero and variance σ^2 is known as **Gaussian white noise**. We write this model as

$$\epsilon_{1:N} \sim \text{iid } N[0, \sigma^2].$$

Example: Binary white noise

Let $\epsilon_{1:N}$ be iid with

$$\epsilon_n = \begin{cases} 1, & \text{with probability } 1/2 \\ -1, & \text{with probability } 1/2 \end{cases}.$$

We can check that $\mathbb{E}[\epsilon_n] = 0$, $\text{Var}(\epsilon_n) = 1$ and $\text{Cov}(\epsilon_m, \epsilon_n) = 0$ for $m \neq n$. Therefore, $\epsilon_{1:N}$ is white noise.

Similarly, for any $p \in (0, 1)$, we could have

$$\epsilon_n = \begin{cases} (1-p)/p, & \text{with probability } p \\ -1, & \text{with probability } 1-p \end{cases}.$$

类似于伯努利分布的白噪音

Example: Sinusoidal white noise

Let $\epsilon_n = \sin(2\pi nU)$, with a single draw $U \sim \text{Uniform}[0, 1]$ determining the time series model for all $n \in 1 : N$. We will show this is an example of a weakly stationary time series that is not strictly stationary.

Question 3.6. Show that $\epsilon_{1:N}$ is weakly stationary, and is white noise!

$$E[\epsilon_n] = \int_0^1 \sin(2\pi nU) du = \frac{1}{2\pi n} \cos[2\pi nU] \Big|_{U=0}^{U=1} = 0$$

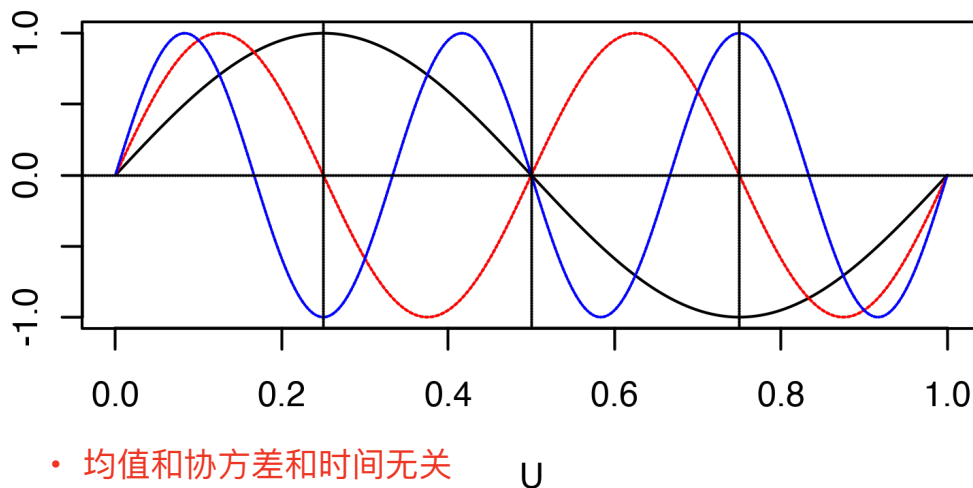
$$\begin{aligned} \text{cov}(\epsilon_m, \epsilon_n) &= E[\epsilon_m \epsilon_n] = \int_0^1 \sin(2\pi mU) \sin(2\pi nU) du \\ &= \frac{1}{2\pi} \int_0^1 \cos 2\pi(m-n)U + \cos 2\pi(m+n)U du \end{aligned}$$

$$\text{当 } m \neq n \text{ 时, } \text{cov} = 0$$

$$\text{当 } m = n \text{ 时 } \text{cov} = \frac{1}{2}$$

Question 3.7. Show that $\epsilon_{1:N}$ is NOT strictly stationary.

Hint: consider the following plot of $\epsilon_{1:3}$ as a function of U . ϵ_1 is shown as the black line; ϵ_2 is the red line; ϵ_3 is the blue line.



- 均值和协方差和时间无关
- 但是具体分布和时间 n 有关

$$P[\epsilon_1=1 | \epsilon_2=0] = 1$$

$$P[\epsilon_3=0 | \epsilon_1=1] = 0$$

这个反例说明联合分布和时间 n 有关

Using white noise to build other time series models

Reminder: Why do we even need time series models?

- All statistical tests (i.e., whenever we use data to answer a question) rely on having a model for the data. The model is sometimes called the **assumptions** for the test.
- If our model is wrong, then any conclusions drawn from it may be wrong. Our error could be small and insignificant, or disastrous.
- Time series data collected close in time are often more similar than a model with iid variation would predict. We need models that have this property, and we must work out how to test interesting hypotheses for these models.

需要一层层证明assumption是正确的

The AR(p) autoregressive model

- The order p autoregressive model, abbreviated to AR(p), is

$$[M1] \quad Y_n = \phi_1 Y_{n-1} + \phi_2 Y_{n-2} + \cdots + \phi_p Y_{n-p} + \epsilon_n,$$

where $\{\epsilon_n\}$ is a white noise process.

是前 p 个数据点的线性组合,
初始化点期望为0则具有平稳性

- Often, we consider the **Gaussian AR(p)** model, where $\{\epsilon_n\}$ is a Gaussian white noise process.
- M1 is a **stochastic difference equation**. It is a difference equation (also known as a recurrence relation) since each time point is specified recursively in terms of previous time points. Stochastic just means random.

如果模型有效,则通过 Y_n 可以递归得到之前的时序数据

- To complete the model, we need to **initialize** the solution to the stochastic difference equation. Supposing we want to specify a distribution for $Y_{1:N}$, we have some choices in how to **set up the initial values**.
- ① We can specify $Y_{1:p}$ explicitly, to get the recursion started.
- ② We can specify $Y_{1-p:0}$ explicitly. 需要p个参数初始化模型
- ③ For either of these choices, we can define these initial values either to be additional parameters in the model (i.e., not random) or to be specified random variables.
- ④ If we want our model is strictly stationary, we must initialize so that $Y_{1:p}$ have the proper joint distribution for this stationary model.
- Assuming the initialization has mean zero, M1 implies that $\mathbb{E}[Y_n] = 0$ for all n . For additional generality, we could add a constant mean μ .
- Let's investigate a particular Gaussian AR(1) process, as an exercise.

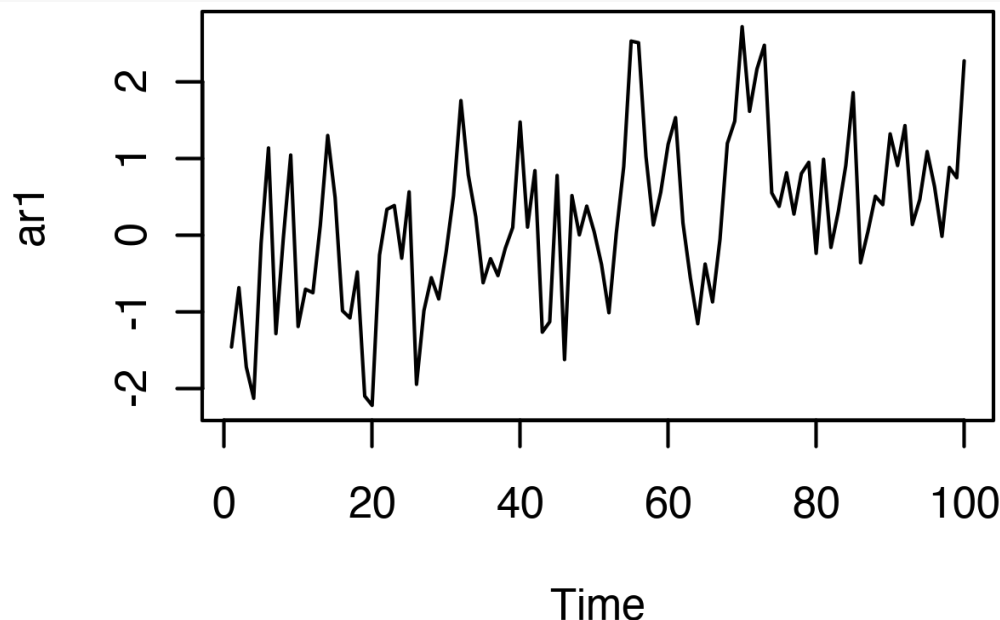
$$[M2] \quad Y_n = 0.6Y_{n-1} + \epsilon_n,$$

where $\epsilon_n \sim \text{iid } N[0, 1]$. We will initialize with $Y_1 \sim N[0, 1.56]$.

Simulating an autoregressive model

Looking at simulated sample paths is a good way to get some intuition about a random process model. We will do this for the AR(1) model M2. One approach is to use the `arima.sim` function in R.

```
set.seed(123456789)
ar1 <- arima.sim(list(ar=0.6),n=100,sd=1)
plot(ar1,type="l")
```

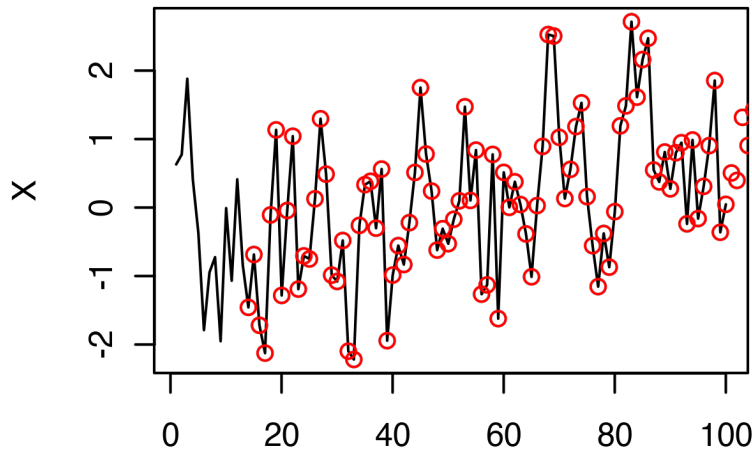


- Does your intuition tell you that these simulated data are evidence for a model with a linear trend?
- The eye looks for patterns in data, and often finds them even when there is no strong statistical evidence.
- That is why we need statistical tests!
- It is easy to see patterns even in white noise. Dependent models produce spurious patterns even more often.
- Play with simulating different models with different seeds to train your intuition.

看起来好像有趋势,但实际上模型是平稳的

We can also simulate model M2 directly by writing the model equation:

```
set.seed(123456789)
N <- 100
X <- numeric(N)
X[1] <- rnorm(1, sd=sqrt(1.56))
for(n in 2:N) X[n] <- 0.6 * X[n-1] + rnorm(1)
plot(X, type="l")
points(lag(ar1, -13), col="red", pch=1)
```



Burn-in to converge to the stationary distribution

Our simulation matches `arima.sim` perfectly with a shift of 13 time points. Explain this. Hint: How does `arima.sim` initialize the simulation?

Question 3.8. What are the advantages and disadvantages of using `arma.sim` over the direct simulation method?

Advantage:

- Efficient in run time
- We write less code and few bugs
- Consistency

Disadvantage:

- Less control
- Easier to debug
- Self-education, requires more understanding

Question 3.9. Compute the autocovariance function for model M2.

用递归计算

$$\text{VAR}[Y_n] = \text{VAR}[0.6Y_{n-1} + \epsilon] = 0.36 \text{VAR}[Y_{n-1}] + \text{VAR}[\epsilon] + \underbrace{\text{COV}}_{=0}$$

基于稳定性

$$\text{VAR}[Y_n] = 0.36 \text{VAR}[Y_n] + 6^2$$

$$\text{VAR}[Y_n] = \frac{6^2}{0.64}$$

$$\begin{aligned} \gamma_h &= \text{COV}[Y_n, Y_{n+h}] = \text{COV}[Y_n, 0.6Y_{n+h-1} + \epsilon_{n+h-1}] \\ &= 0.6 \text{COV}[Y_n, Y_{n+h-1}] = 0.6 \gamma_{h-1} \\ \gamma_h &= \frac{6^2}{0.64} \times 0.6^h \end{aligned}$$

The MA(q) moving average model

- The order q moving average model, abbreviated to MA(q), is

$$[M3] \quad Y_n = \epsilon_n + \theta_1 \epsilon_{n-1} + \cdots + \theta_q \epsilon_{n-q},$$

where $\{\epsilon_n\}$ is a white noise process. 和之前 q 个点的噪音有关

- To fully specify $Y_{1:N}$ we must specify the joint distribution of $\epsilon_{1-q:N}$.
- Often, we consider the **Gaussian MA(q)** model, where $\{\epsilon_n\}$ is a Gaussian white noise process.
- In M3, we've defined a zero mean process. We could add a mean μ .
- Let's investigate a particular Gaussian MA(2) process, as an exercise.

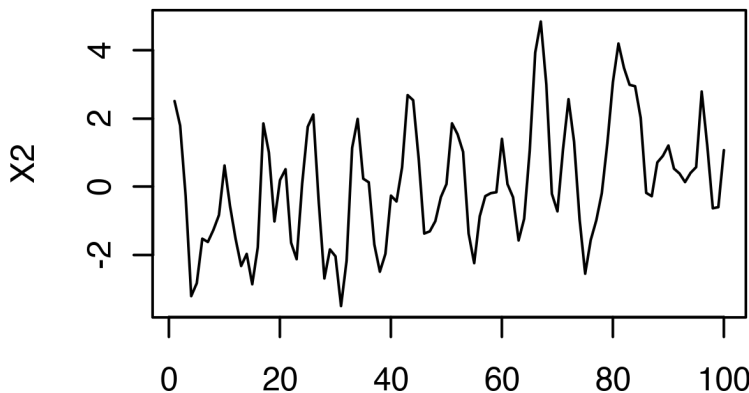
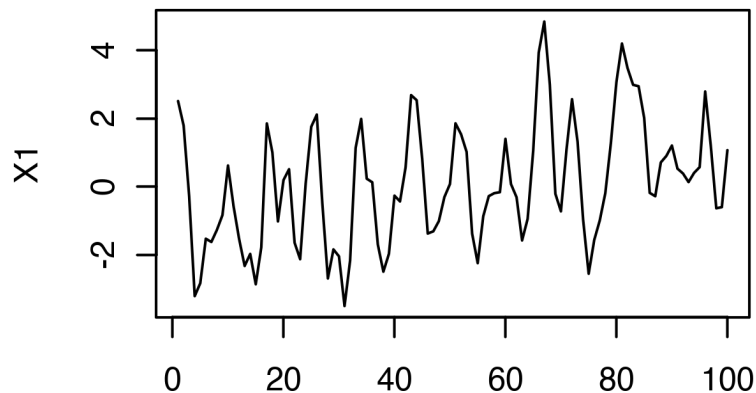
$$[M4] \quad Y_n = \epsilon_n + 1.5\epsilon_{n-1} + \epsilon_{n-2},$$

where $\epsilon_n \sim \text{iid } N[0, 1]$.

Simulating a moving average model

We simulate M4 using the same methods as for the autoregressive model.

```
N <- 100
set.seed(123456789)
X1 <- arima.sim(list(ma=c(1.5,1)),n=N,sd=1)
set.seed(123456789)
epsilon <- rnorm(N+2)
X2 <- numeric(N)
for(n in 1:N) X2[n] <- epsilon[n+2]+1.5*epsilon[n+1]+epsilon[n]
plot(X1,type="l") ; plot(X2,type="l")
```



X1 and X2 look identical. We can check this

```
all(X1==X2)
```

```
[1] TRUE
```

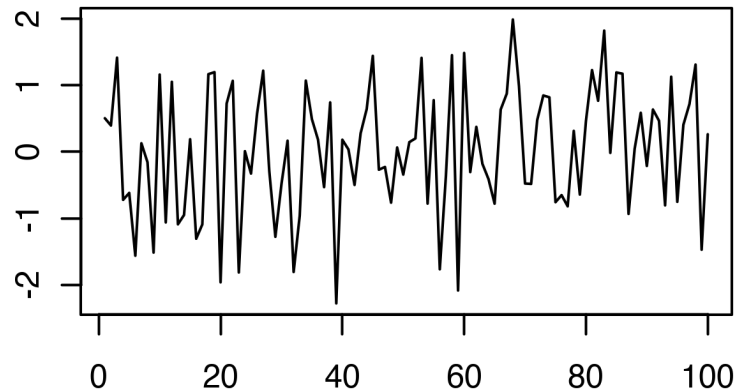
Do you agree that the spurious evidence for a trend that we saw for the AR(1) model is still somewhat present for the MA(2) simulation? Let's see if we can also see it in the underlying white noise process:

```
N <- 100
```

```
set.seed(123456789)
```

```
epsilon <- rnorm(N)
```

```
plot(epsilon,type="l")
```



The random walk model

- The **random walk** model is

$$[M5] \quad Y_n = Y_{n-1} + \epsilon_n,$$

where $\{\epsilon_n\}$ is white noise. Unless otherwise specified, we usually initialize with $Y_0 = 0$.

- If $\{\epsilon_n\}$ is Gaussian white noise, then we have a Gaussian random walk.
- The random walk model is a special case of AR(1) with $\phi_1 = 1$.
- The stochastic difference equation in M5 has an exact solution,

$$Y_n = \sum_{k=1}^n \epsilon_k.$$

- We can also call $Y_{0:N}$ an **integrated white noise process**. We think of summation as a discrete version of integration.
- If data $y_{1:N}^*$ are modeled as a random walk, the value of Y_0 is usually an unknown. Rather than introducing an unknown parameter to our model, we may initialize our model at time t_1 with $Y_1 = y_1^*$.

- The first difference time series $z_{2:N}$ is defined by

$$z_n = \underline{\Delta y_n^*} = y_n^* - y_{n-1}^* \quad \text{一阶差分} \quad (3)$$

- From a time series of length N , we only get $N - 1$ first differences.
- A random walk model for $y_{1:N}^*$ is essentially equivalent to a white noise model for $z_{2:N} = \Delta y_{2:N}^*$, apart from the issue of initialization.

The random walk with drift

- The **random walk with drift** model is given by the difference equation

$$[M6] \quad Y_n = Y_{n-1} + \mu + \epsilon_n,$$

driven by a white noise process $\{\epsilon_n\}$. This has solution

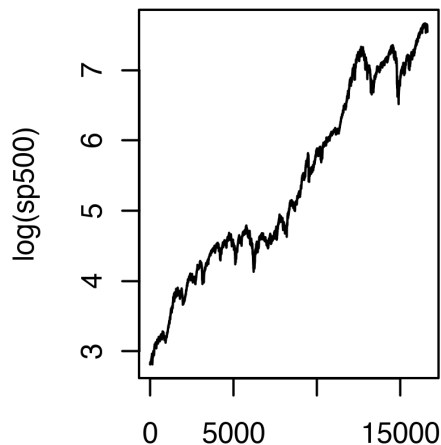
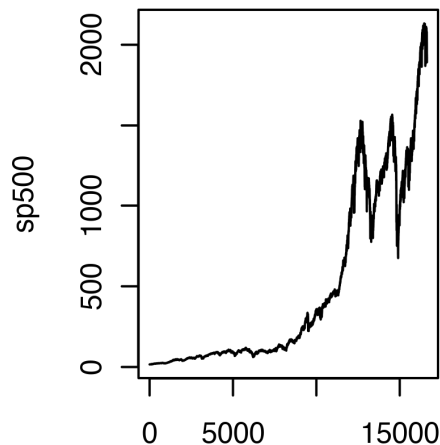
$$Y_n = Y_0 + n\mu + \sum_{k=1}^n \epsilon_k. \quad \text{每次添加了一个常数}$$

- Here, μ is the mean of the **increments** rather than the random walk process itself.
- As for the random walk without drift, we must define Y_0 to initialize the model and complete the model specification. Unless otherwise specified, we usually initialize with $Y_0 = 0$.

Modeling financial markets as a random walk

The theory of **efficient financial markets** suggests that the logarithm of a stock market index (or the value of an individual stock, or other investment) might behave like a random walk with drift. We test this on daily S&P 500 data, downloaded from yahoo.com. 需要有效市场假说

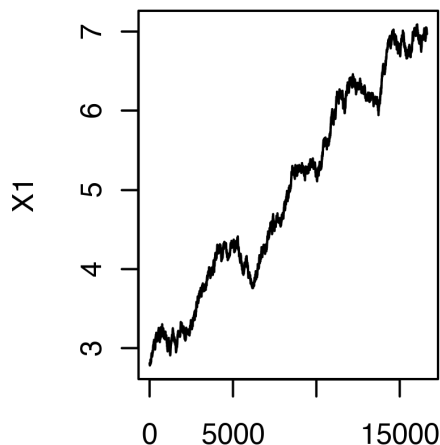
```
dat <- read.table("sp500.csv", sep=",", header=TRUE)
N <- nrow(dat)
sp500 <- dat$Close[N:1] # data are in reverse order in sp500.csv
par(mfrow=c(1,2))
plot(sp500, type="l") ; plot(log(sp500), type="l")
```



To train our intuition, we compare the data with simulations from a fitted model. A simple starting point is a Gaussian random walk with drift, having parameters estimated from the data.

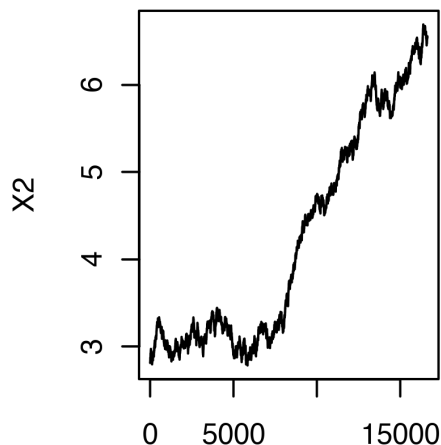
```
mu <- mean(diff(log(sp500)))
sigma <- sd(diff(log(sp500)))
X1 <- log(sp500[1])+cumsum(c(0,rnorm(N-1,mean=mu,sd=sigma)))
X2 <- log(sp500[1])+cumsum(c(0,rnorm(N-1,mean=mu,sd=sigma)))
plot(X1,type="l") ; plot(X2,type="l")
```

计算差分的均值和标准差来估计模型



Index

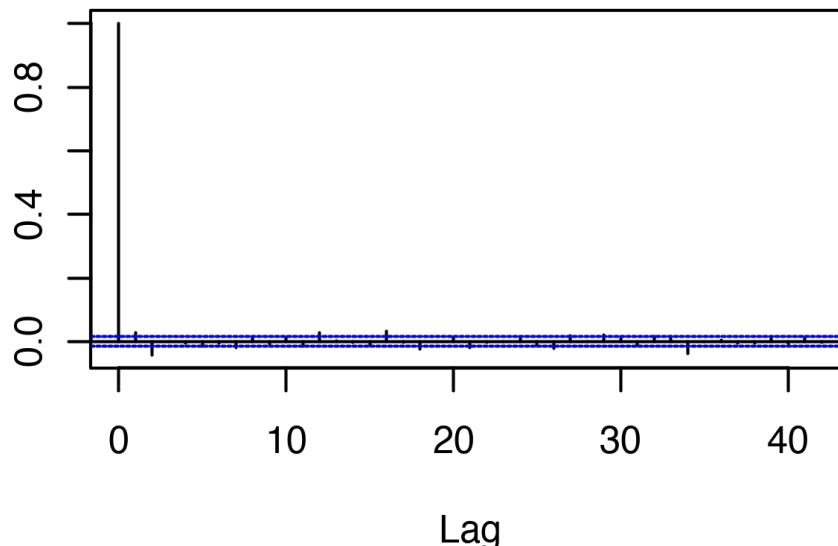
Just different simulation



Index

- This seems reasonable so far. Now we plot the sample autocorrelation function (sample ACF) of `diff(log(sp500))`.
- It is bad style to refer to quantities using computer code notation. We should set up mathematical notation in the text. Let's try again...
- Let $y_{1:N}^*$ be the time series of S&P 500 daily closing values downloaded from yahoo.com. Let $z_n = \Delta \log y_n^* = \log y_n^* - \log y_{n-1}^*$.
- We plot the sample autocorrelation function of the time series of S&P 500 returns, $z_{2:N}$.

```
z <- diff(log(sp500))
acf(z)
```



- This looks close to the ACF of white noise. There is some evidence for a small nonzero autocorrelation at lags 0 and 1.
- Here, we have a long time series ($N = 16616$). For such a long time series, statistically significant effects may be practically insignificant.

Question 3.10. Why may the length of the time series be relevant when considering practical versus statistical significance?

随着length增加,置信区间会变小,这时候
small sample will be significant

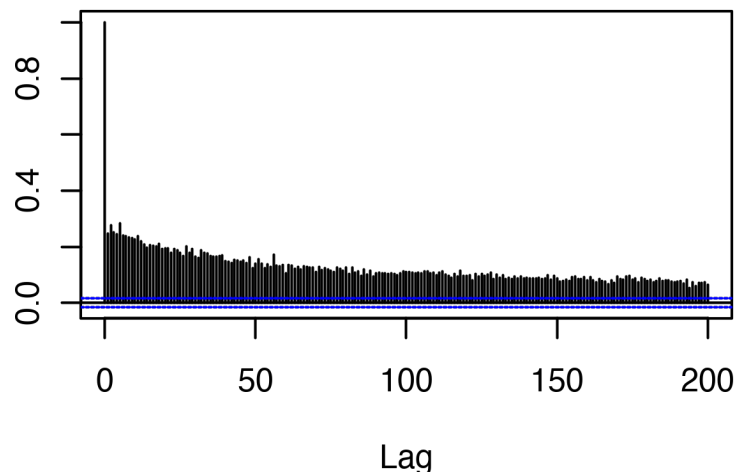
- It seems like the S&P 500 returns (centered, by subtracting the sample mean) may be a real-life time series well modeled by white noise.

- Looking at the absolute value of the centered returns is thought-provoking.

Question 3.11. How should we interpret the following plot? To what extent does this plot refute the white noise model for the centered returns (or, equivalently, the random walk model for the log index value)?

Volatility is persistent, 剧烈波动往往会延续到下个时间点

```
acf(abs(z-mean(z)), lag.max=200)
```



Volatility and market inefficiencies

- Nowadays, nobody is surprised that the sample ACF of a financial return time series shows little or no evidence for autocorrelation.
- Deviations from the efficient market hypothesis, if you can find them, are of interest.
- Also, it remains a challenge to find good models for **volatility**, the conditional variance process of a financial return model.

Further reading

- Chapter 1 of Shumway and Stoffer (2017) provides a complementary introduction to time series analysis.
- If you are relatively new to R, Venables *et al.* (2021) is a comprehensive introduction.

References and Acknowledgements

Shumway RH, Stoffer DS (2017). *Time Series Analysis and its Applications: With R Examples*. Springer. URL <http://www.stat.pitt.edu/stoffer/tsa4/tsa4.pdf>.

Venables WN, Smith DM, Team RDC (2021). “An introduction to R (Version 4.1.2).” URL <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>.

- Compiled on January 10, 2022 using R version 4.1.2.
- Licensed under the [Creative Commons Attribution-NonCommercial](#) license. Please share and remix non-commercially, mentioning its origin.
- We acknowledge [previous versions of this course](#).

